Nuclear Breakup in eA
The Fate of the Nucleus

Thomas Ullrich

BNL

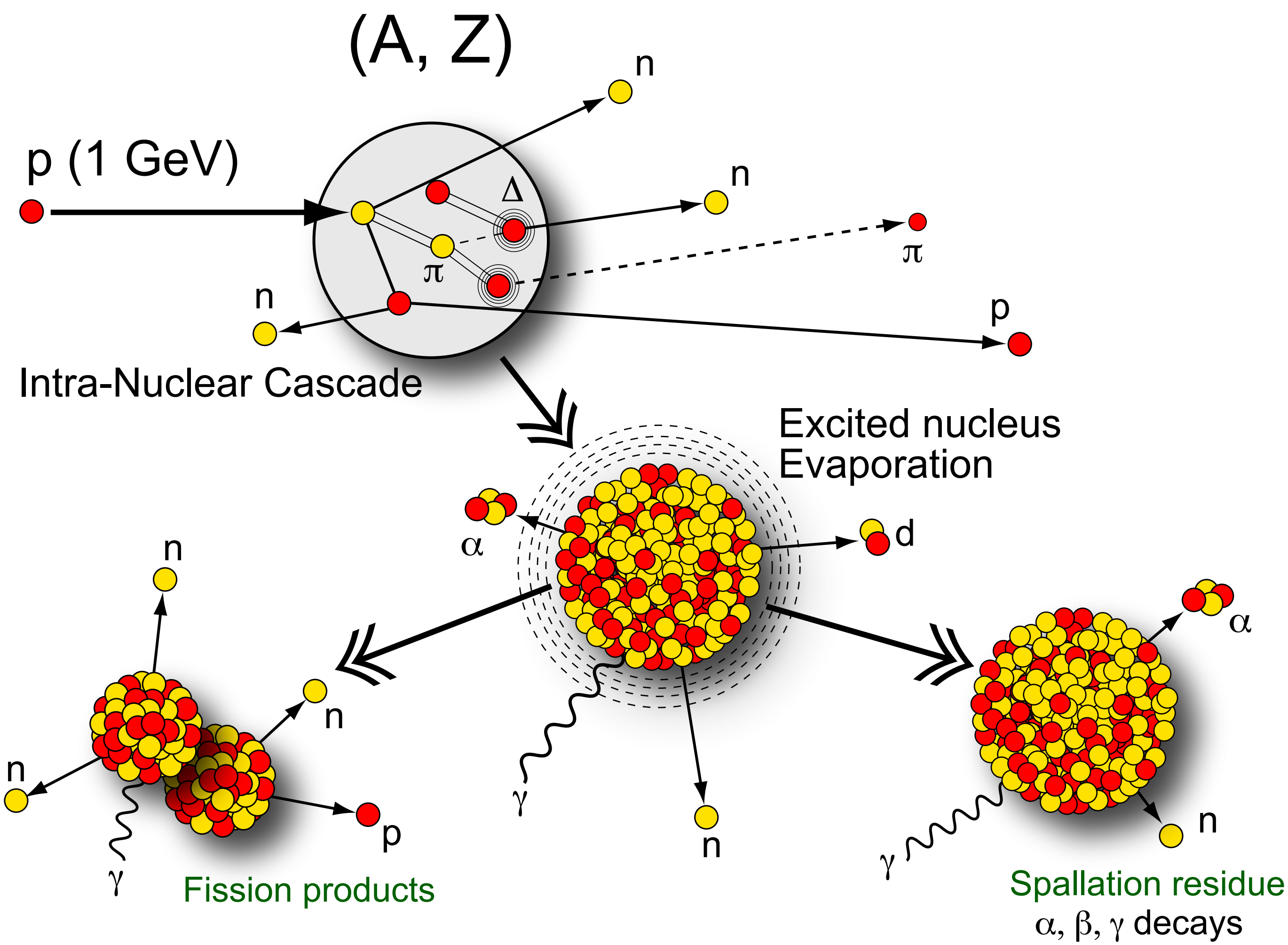
December 9, 2010

We are not alone ...

The issue of nuclear breakup is of great importance for several communities

- Understanding the space environment
  - Radiation damage
- Nuclear waste disposal
  - Transmutation of waste (ATW) systems
- Nuclear spallation
- Cosmic ray showers
- HEP/NP detector simulations

Physical Picture (h+A case)



Intra-Nuclear Cascade

- Particle production
- Remnant Nucleus (A, Z, E*, ...)

De-Excitation

- Evaporation
- Fission
- Residual Nuclei

Basic idea is that after the cascade the excited nucleus lost all memory of how it got there and all that matters is A, Z, E*.

De-Excitation Models (Statistical Models)

SMM (Statistical Multifragmentation Model)

- simultaneous breakup
  - thermodynamical configuration weights
  - remnant slits in several chunks
- fragment de-excitation
  - Fermi break-up
  - evaporation $Z \leq 2$ (Weisskopf Ewing)
  - fission (Bohr-Wheeler)
- Bondorf et al.

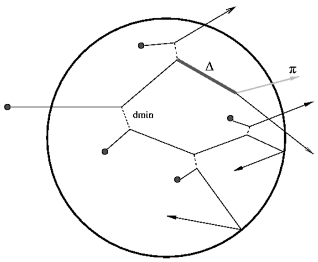
GEMINI++

- no simultaneous break-up
- sequence of binary decays
- evaporation $Z \leq 3$ (Hauser-Feshbach)
- asymmetric fission $Z > 3$ (Moretto)
- symmetric fission (Bohr-Wheeler)
- R. Charity (Charity, 2008), http://www.chemistry.wustl.edu/~rc/gemini++/

Old GEMINI has shortcomings for large nuclei – also cannot describe multifrag. data (due to seq. breakup). Has higher prob. for light fragments (epecially n)

ABLA, MMM, MMMC, SIMON, ISMM, Dresner, GEM, EES, HIFGM, ...

Intra-Nuclear Cascade Model

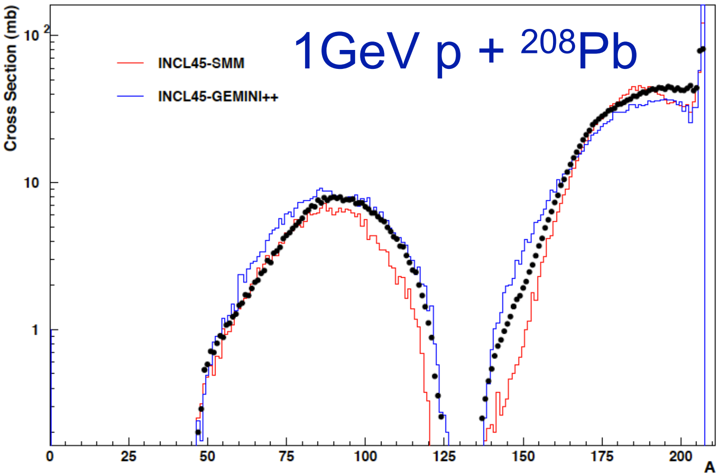


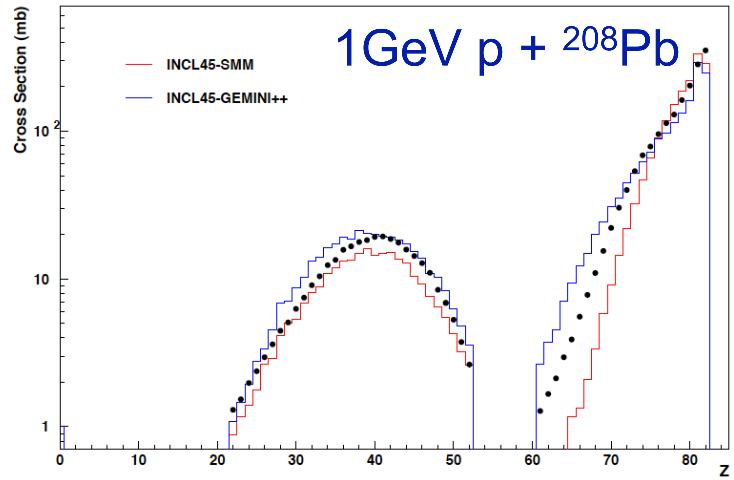
- ISABEL
- INCL4 (state of the art?)
  - typically as combination INCL4/ABLA
  - developed by ULg@Liège, CEA@Saclay
  - binary nucleon-nucleon collisions
  - Nucleus (remnant) left in excited state
  - Must be coupled to a de-excitation code
- For AA typical candidates were: QMD, BUU, VUU

These are typically at lower energies

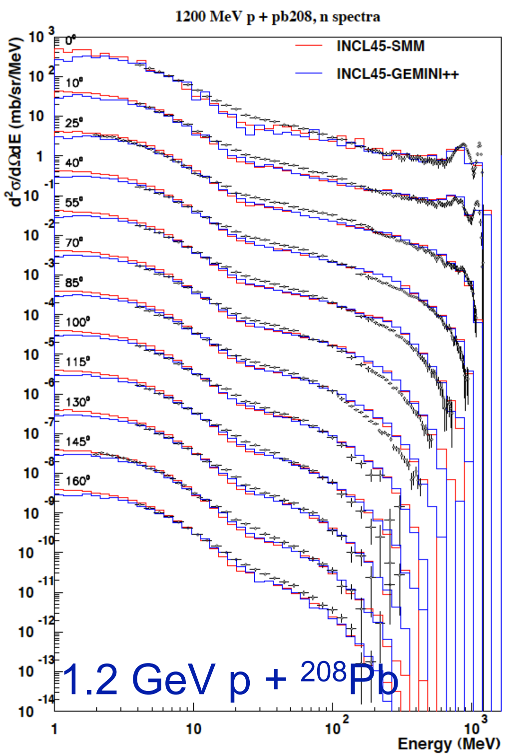
Issue: Time until “statistical” (de-excitation) models are applicable (typical 100 fm/c - no problem for us)

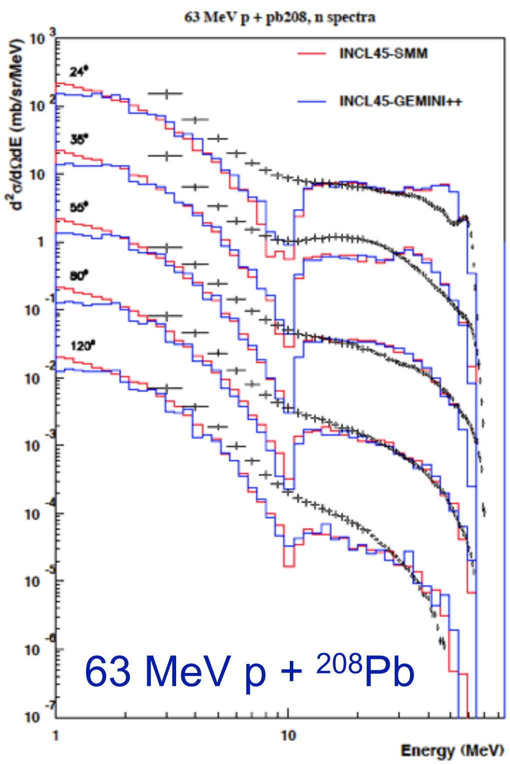
GEMINI++ vs. SMM ()

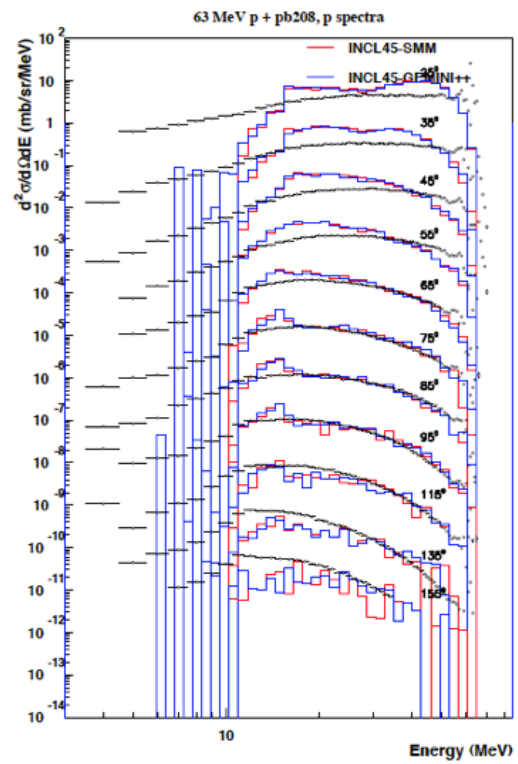




Gemini is pretty accurate







INCL4-ABLA Combination

Often named combination: INCL4-ABLA

(others are: Bertini-Dresner)

Examples:

- included in MCNPX (MCNPX is a general-purpose Monte Carlo radiation transport code for modeling the interaction of radiation with everything.) https://mcnpx.lanl.gov/
- Attempts to put it in GEANT4

Often heard: Replacing ABLA with Gemini++ improves comparison with data

DPMJet-II/DTUNUC

Ansatz for h-A collisions (same as usual)

- A formation zone intra-nuclear cascade of low energy secondaries inside the target nucleus is discussed
- Calculate excitation energies of residual nuclei left after the intra-nuclear cascade process
- treat their further disintegration by introducing models for the evaporation of protons, neutrons, and light fragments, high energy fission, and by applying a Fermi Break-up model to light nuclear fragments.

home-made chain: intra-nuclear cascades → de-excitations

While the intra-cuclear cascades at high energies justifies all the efforts put into that part, it is beyond me while for the de-excitation part they reinvented the wheel (and far less sophisticated as modern tools such as GEMINI, SMM, or ALBA)

Ref: A. Ferrari et al., nucl-th/9509039 (2005) - lots of comparison with data

FLUKA

A. Ferrari who worked on the cascade/evaporation part is also member of the FLUKA team

From the manual:

16.15.2 Cascade-Preequilibrium model (PEANUT)

developed by Ferrari and Sala: It combines an intranuclear part and a preequilibrium part

16.15.3 Evaporation/Fission and Fermi Break-Up

A completely new evaporation treatment was developed by Ferrari and Sala in 1996 and 1997 in substitution of the improved Dresner model.

DPMjet-III and Mods by Nestor/Fluka

DPMjet-III:

- A Monte Carlo event generator for high-energy hadron-hadron, hadron-nucleus, nucleus-nucleus and photon-nucleus collisions
- Particle production in the fragmentation region(s) of the participating nucleus (nuclei) is described by a formation zone suppressed intranuclear cascade followed by Monte Carlo realizations of models for evaporation processes of light nucleons and nuclei, high-energy fission, spectator fragmentation (so far limited to light spectator nuclei) and de-excitation of residual nuclei by photon emission.
- DIS off nuclei is simulated by LEPTO followed by the full intra-nuclear cascade and fragmentation treatment as mentioned above.
- I assume that DPMJet-III contains rather old evaporation code by Ferrari that was later improved in FLUKA only (by Ferrari)

Nestor's version of DPMJet for LHeC uses FLUKA for evaporation in

Geant4

Geant4 included at some point FLUKA but got into license trouble so Geant4-Fluka was removed. Geant4 re-invented the wheel again multiple times:

Physics Reference Manual Version: geant4 9.2 (19 December, 2008):

This model is based on a re-engineering of the INUCL code and includes the Bertini intra-nuclear cascade model with excitons, a pre-equilibrium model, a nucleus explosion model, a fission model, and an evaporation model. In- termediate energy nuclear reactions from 100 MeV to 5 GeV are treated for protons, neutrons, pions, photons and nuclear isotopes.

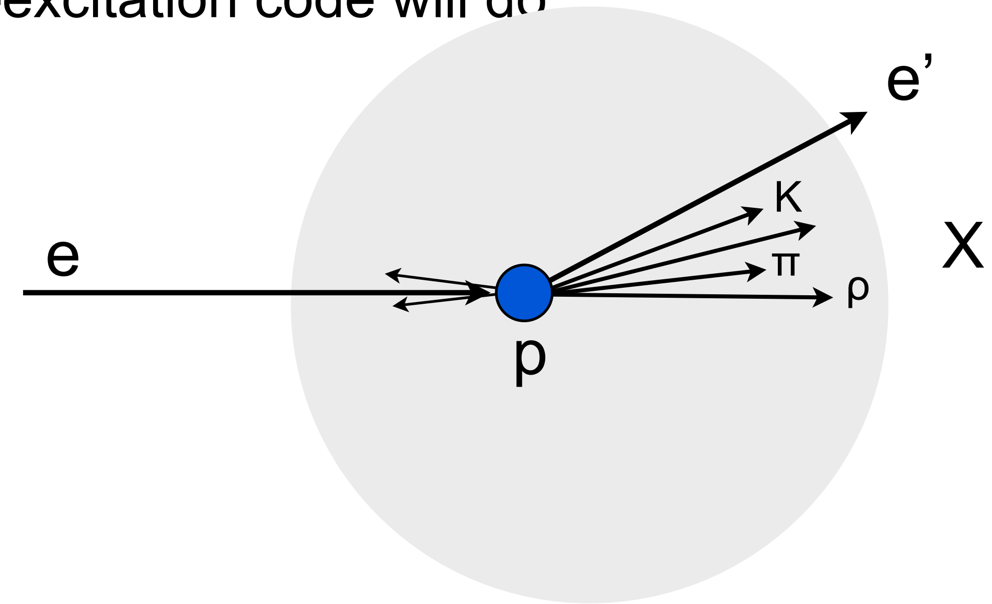
25 Bertini Intranuclear Cascade Model in Geant4 (re-engineered INUCL code and Bertini intra-nuclear cascade model
26 The Geant4 Binary Cascade (transport in nuclei)
27 Abrasion-ablation Model (as binary cascade, faster but less accurate)
28 Electromagnetic Dissociation Model
29 Precompound model.
30 Evaporation Model (uses GEM - not by default)
31 Fission model.
32 Fermi break-up model.
33 Multifragmentation model.
34 INCL 4.2 Cascade and ABLA V3 Evaporation with Fission (alternative or addition?)

What do we really need for eA?

DIS

- Here the critical part is the “intra-nuclear” cascade
  - code exist only for p,n,K,π + A
  - mostly the energy range of incident hadron too low
  - any reasonable de-excitation code will do

- Use e+p generator
- Generated particles become seed for intra-nuclear cascade handled by ripped out FLUKA or Geant4 code



- Issues
  - intra nuclear cascade needs to start **in** the nucleus while G4 and FLUKA are not designed this way (meant for incident h on material)
  - Multi-particle cascade at input (not designed this way)

What do we really need for eA?

- Possibility B:
  - use DPMJet-III (Nestor version) as is (based on Lepto)
  - does more or less already what (A) suggests
  - might even add radiative corrections
  - can switch from FLUKA to any more modern evaporation code in future - but for now FLUKA might do
  - Issues:
    - no support (Stefan Roessler is out of business for now)
    - Very hard to read F77 code - not easy to modify
    - endless licensing issues with FLUKA

What do we really need for eA?

Incoherent Diffractive Events

- Intermediate Regime (absent with protons)
  - the nucleus breaks up into its constituents nucleons, intermediate |t|
  - no intra-nuclear cascade
  - use de-excitation code (e.g. GEMINI++) directly as is with $E^* = \sqrt{-t}$
- Fully Incoherent Diffraction
  - the nucleons undergo inelastic scattering, dominates at large |t|
  - One possibility would to use **`epsoft`** from ZEUS that handles incoherent ep breakup and use those to start intra-nuclear cascade + evaporation/de-excitation
  - Same issues here as for DIS - no idea how to do this?
  - Note cannot use DPMJet-III here (no diffraction) ...

First Look at GEMINI++

Comes in two flavors:

- gemini++: complete code + examples
- geminiRoot: same as above but with hooks to run within ROOT (ClassDef's etc.) - ignore

First steps:

- make
- run example testDecay: **`*** Break *** segmentation violation`**
- Code has some problems
- Needs some overhaul
    - code violates at times ISO C++ rule
    - many warnings
    - for experts:
        - massive use of static data member that cross-reference each other ⇒ C ++ compiler doesn't guarantee any order of instantiation of static data
        - slow performance since table/file I/O done in constructors that are called over and over again (reading the same content)
        - solution: get rid of static member and use singletons or plain functions instead

First Look at GEMINI++

```
void CNucleus::setCompoundNucleus(float fEx0, float fJ0)
{
  excite(fEx0,fJ0);
  origin = 0;
  timeSinceStart = 0.;
  sumGammaEnergy = 0.;
  iPoint = -1;
}
void CNucleus::setCompoundNucleus(double dEx0, float fJ0)
{
  setCompoundNucleus((float)dEx0,fJ0);
}
void CNucleus::setCompoundNucleus(float fEx0, double dJ0)
{
  setCompoundNucleus(fEx0,(float)dJ0);
}
void CNucleus::setCompoundNucleus(double dEx0, double dJ0)
{
  setCompoundNucleus((float)dEx0,(float)dJ0);
}
```

While not a problem makes searching and understanding the code difficult ...

Some things to get used to:

- Units are in MeV
- ID of produced fragments is a name string (not an ID)
- Gamma-rays are there but no idea yet how to get them
  - `float CNucleus::gammaWidth()`
  - `float CNucleus::gammaWidthMultipole(int iMode)`
  - `float CNucleus::gammaWidthE1GDR()`
  - It's a different world ;-)

Modifications to Gemini++

- Removed static variable mess
  - replace by functions or singletons
- Replaced `rand`() by `drand48`()
- Replaced all `float` by `double` (speed up of code)
- added CNucleus: getZ(), getA(), getN()
- added TLorentzVector CNucleus:getLorentzVector()
- Gamma-rays are there but no idea yet how to get them
  - `float CNucleus::gammaWidth()`
  - `float CNucleus::gammaWidthMultipole(int iMode)`
  - `float CNucleus::gammaWidthE1GDR()`
  - It’s a different world ;-)

First Look at GEMINI++

Example event output of my first test program:

event = 4 , E* = 290 MeV

number of products= 21

stable fragment= 4He parent = 197Au

stable fragment= n parent = 193Ir

stable fragment= t parent = 192Ir

stable fragment= n parent = 189Os

stable fragment= n parent = 188Os

stable fragment= n parent = 187Os

stable fragment= n parent = 186Os

stable fragment= n parent = 185Os

stable fragment= 4He parent = 184Os

stable fragment= n parent = 180W

stable fragment= n parent = 179W

stable fragment= n parent = 178W

stable fragment= n parent = 177W

stable fragment= n parent = 176W

stable fragment= n parent = 175W

stable fragment= n parent = 174W

stable fragment= n parent = 173W

stable fragment= n parent = 172W

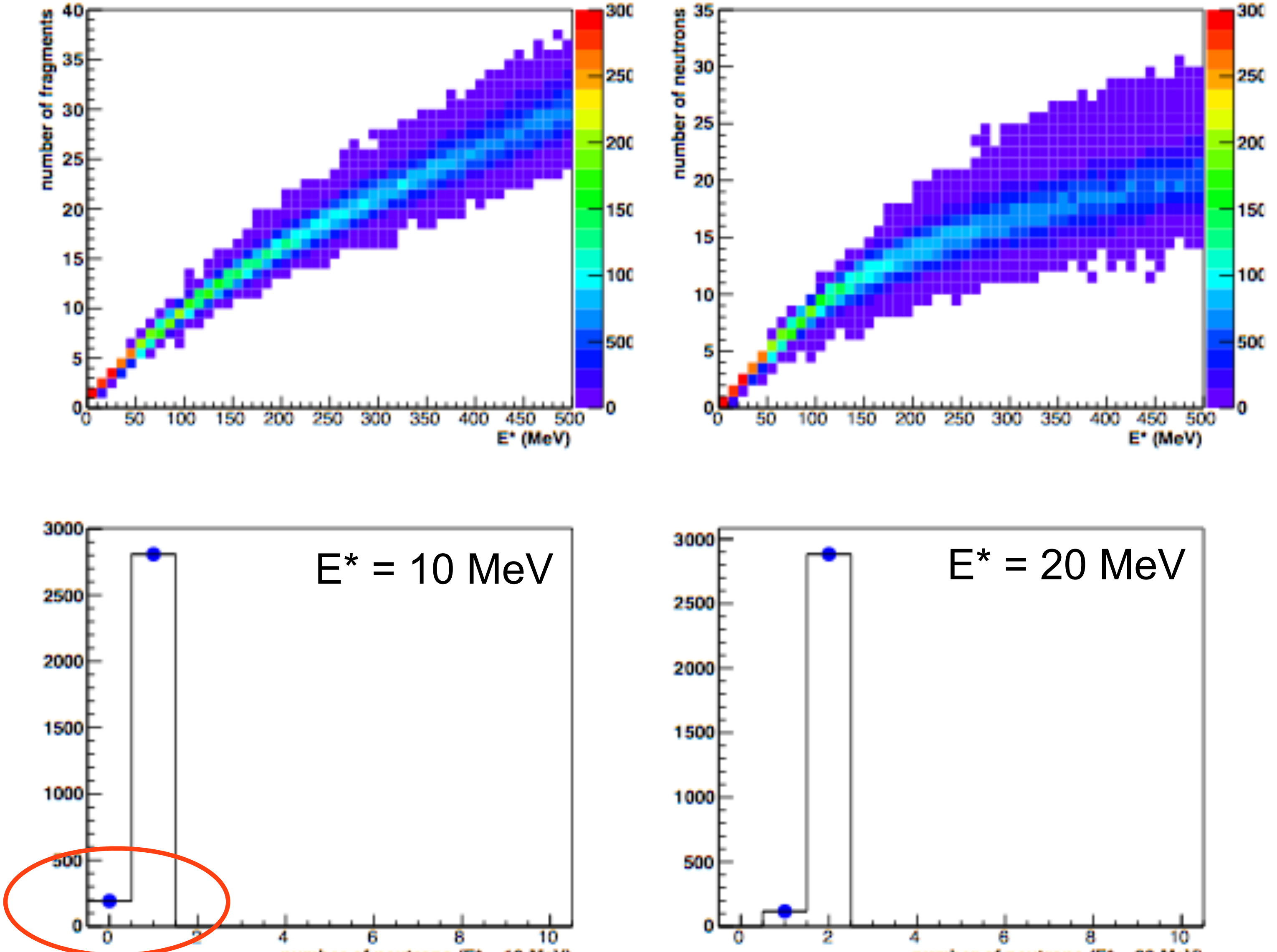
stable fragment= n parent = 171W

stable fragment= n parent = 170W

stable fragment= 169W parent = 170W

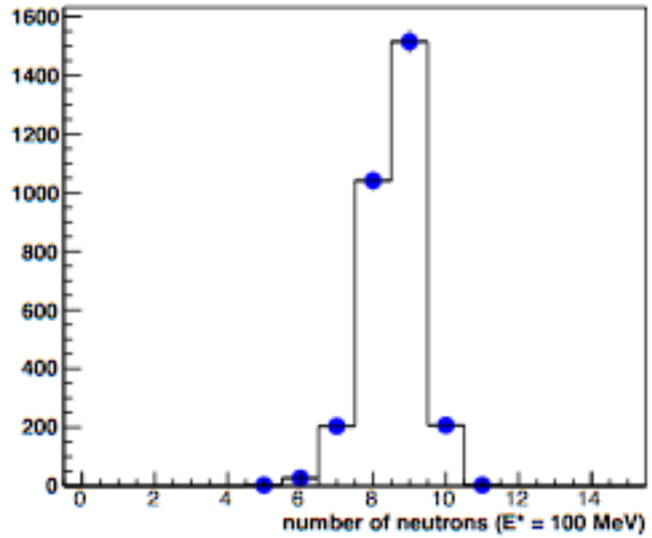
First Look at GEMINI++

all in CM of nucleus (here Au)

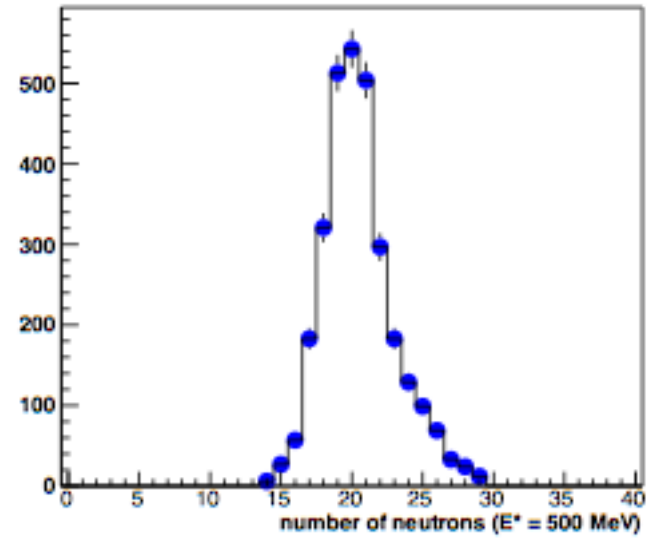


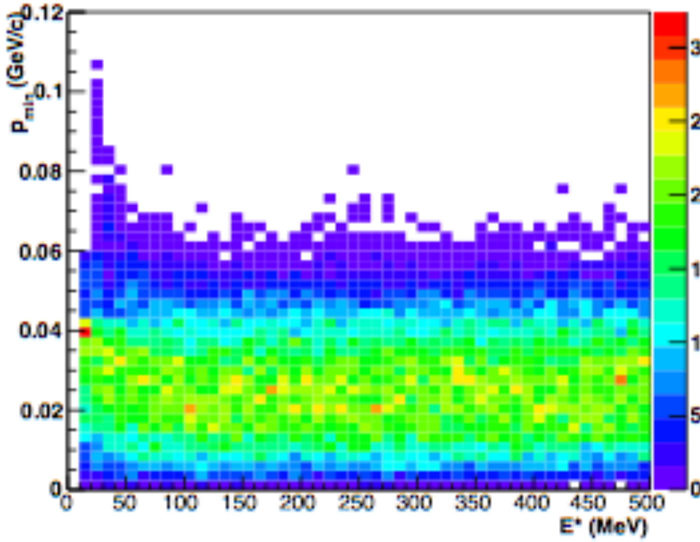
First Look at GEMINI++

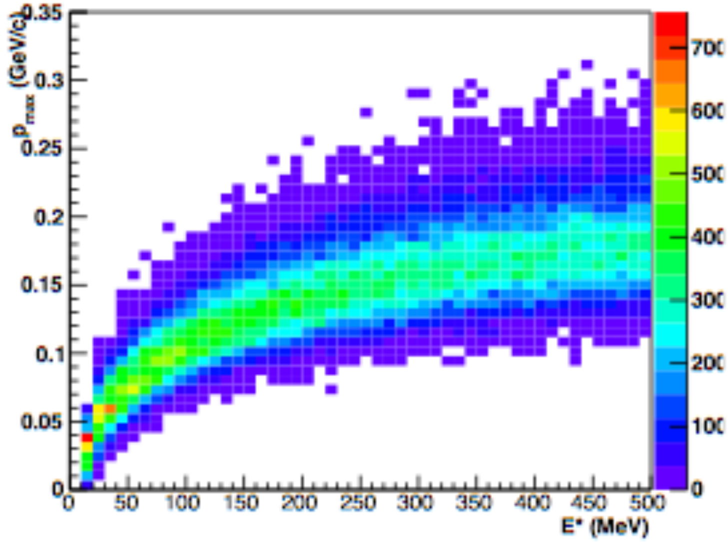
E* = 100 MeV



E* = 500 MeV

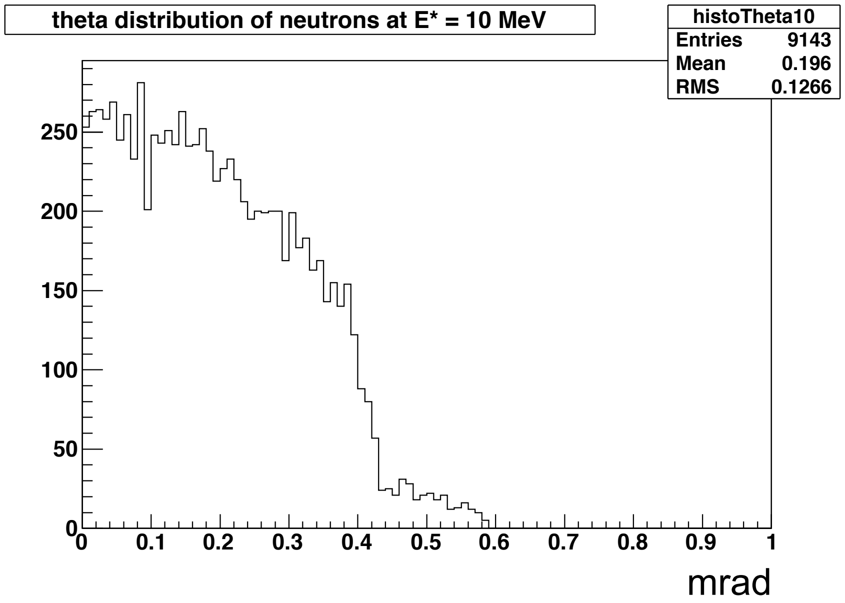


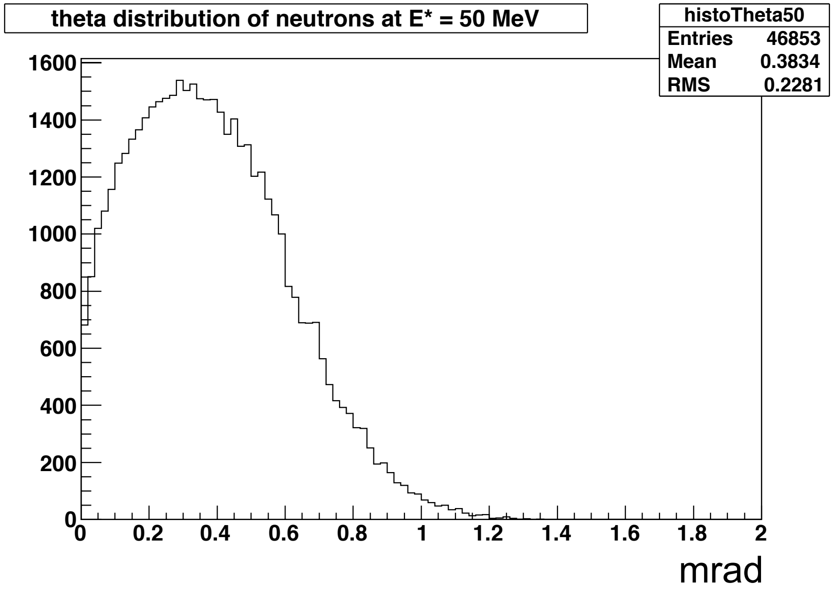


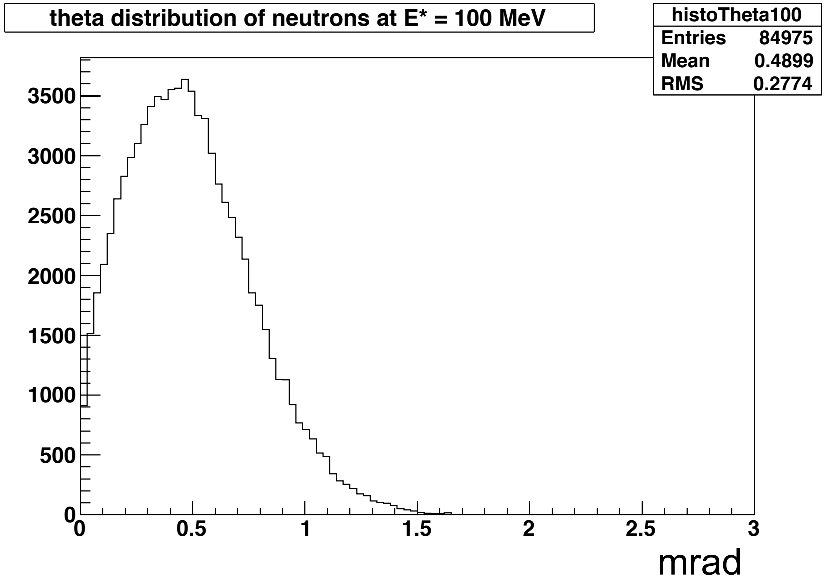


First Look at GEMINI++

100 GeV Au beam

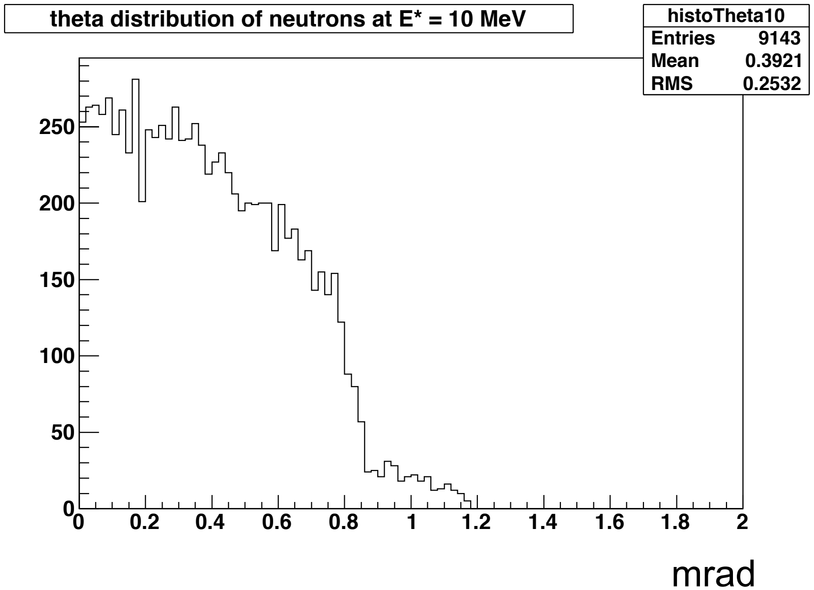


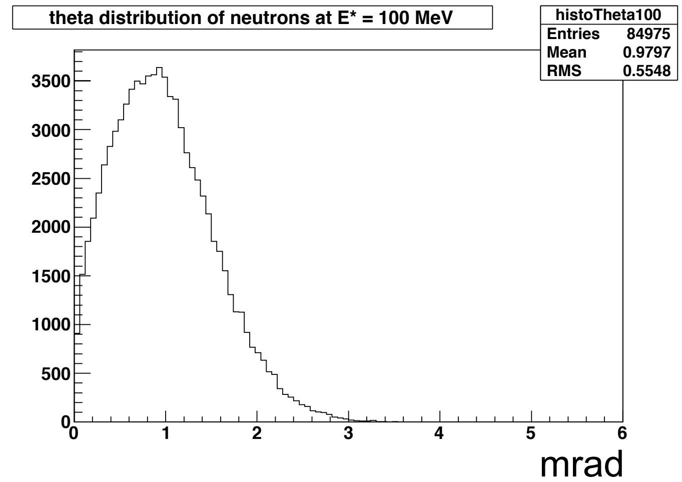


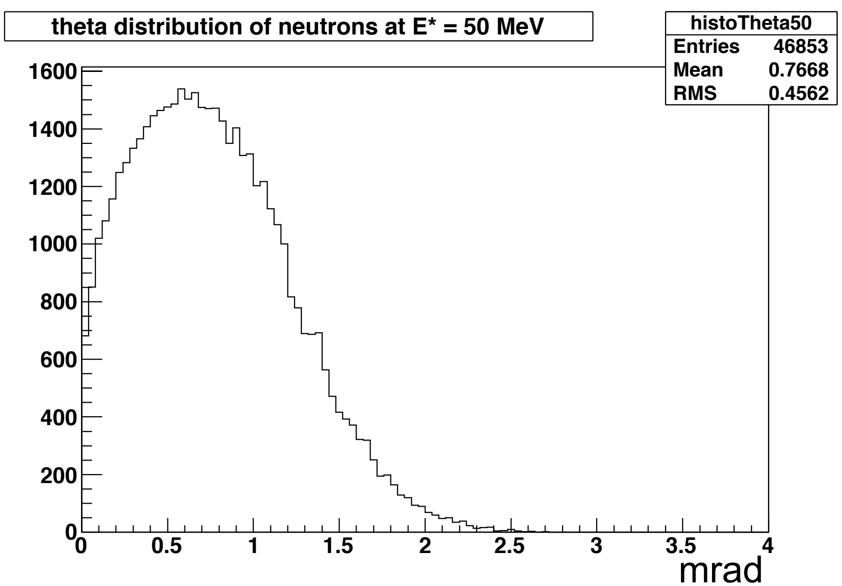


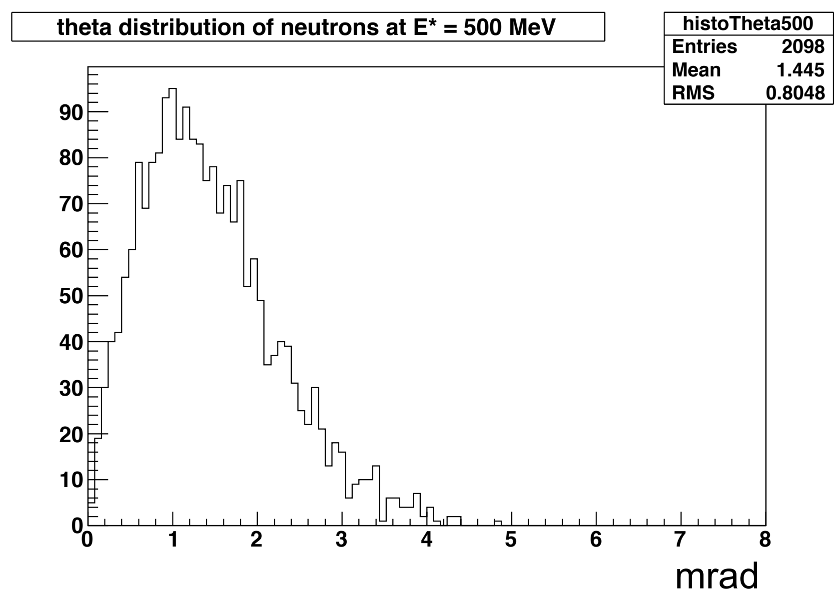
First Look at GEMINI++

50 GeV Au beam









To do ...

Contact NASA folks at BNL (Adam Rusek 911B)

Gemini++:

- photons
- boost into lab frame
- build as usable afterburner

Verify with other simulators

- SMM
- ABLA
- Cons: cost lots of time
- Pro: need to be 100% sure (affects IR design)
- nice job for someone interested in e+A ....

References

- *Statistical description of nuclear breakup*, Botvina and Mishustin, nucl-th/0510081 (2005)
- *Validation of high-energy nuclear models: State-of the art and perspectives*, S. Leray et al., NIM A562, 806 (2006)
- *Comparison of stat. multifragmentation and evaporation models for heavy-ion collisions*, M. Tsang et al., Eur. Phys. J. A 30, 129 (2006)
- CASCADE PARTICLES, NUCLEAR EVAPORATION, AND RESIDUAL NUCLEI IN HIGH ENERGY, HADRON-NUCLEUS INTERACTIONS, A. Ferrari et al. (DPMJet-II), nucl-th/9509039 (2005)
- Geant 4: http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual
- WWW: use “de-excitation” and “intra-nuclear cascade models” in search engine and you get more references than one can handle

Update

December 14, 2010

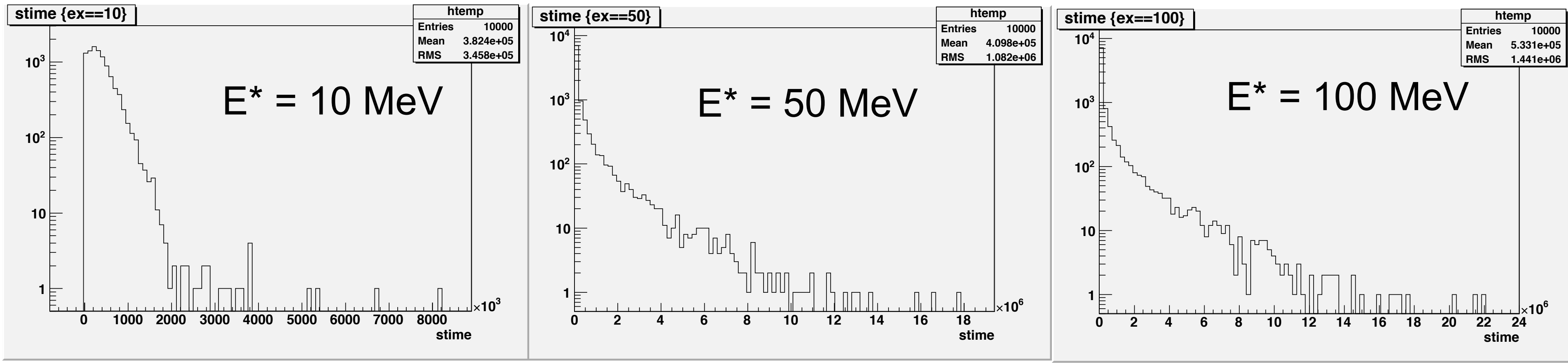
Update Gemini++

Now in contact with Bob Charity who is very helpful:

- photons are not build in and it would take a lot of work to get this properly done. Bob is mainly interested in “particles”
- Bob also got me in contact with the ABLA and SMM guys (both at GSI) who appear to be also helpful and very interested
  - Aleksandra Kelić-Heil (ABLA): A.Kelic@gsi.de
  - Alexander Botvina (SMM): A.Botvina@gsi.de
  - they will send me the code
- I sense that these guys are interested
  - a possible “hook” to make EIC interesting for NS?
  - incoherent diffractive eA is a new way to look at

Emission times

The time is given in units of zs = 10^{-21} seconds and it is the time since the creation of the compound nucleus.

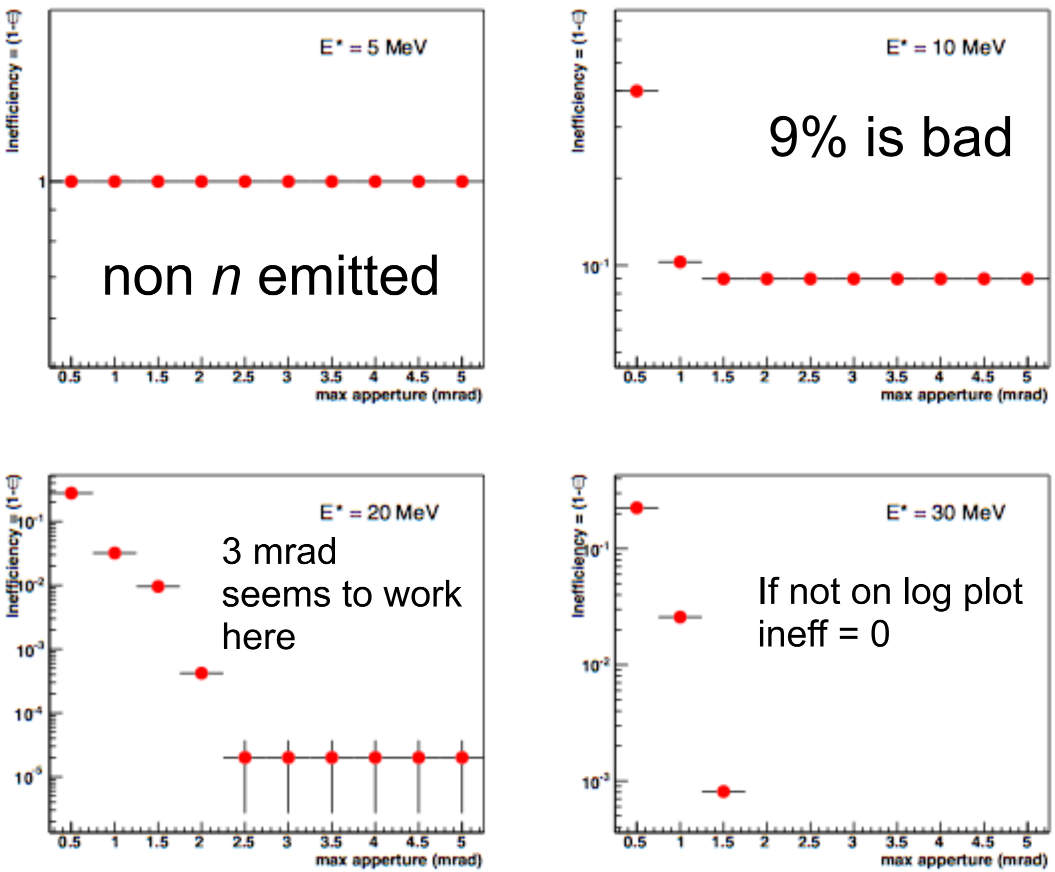


- Absolute longest times observed is 24 fm
- Even with a boost with γ=100 this is not an issue
- ... assuming the time until the compound nucleus is formed is short (~ 100 fm)

Acceptance studies

Here 50 GeV Au beams

- For a given aperture check how many *n* get accepted



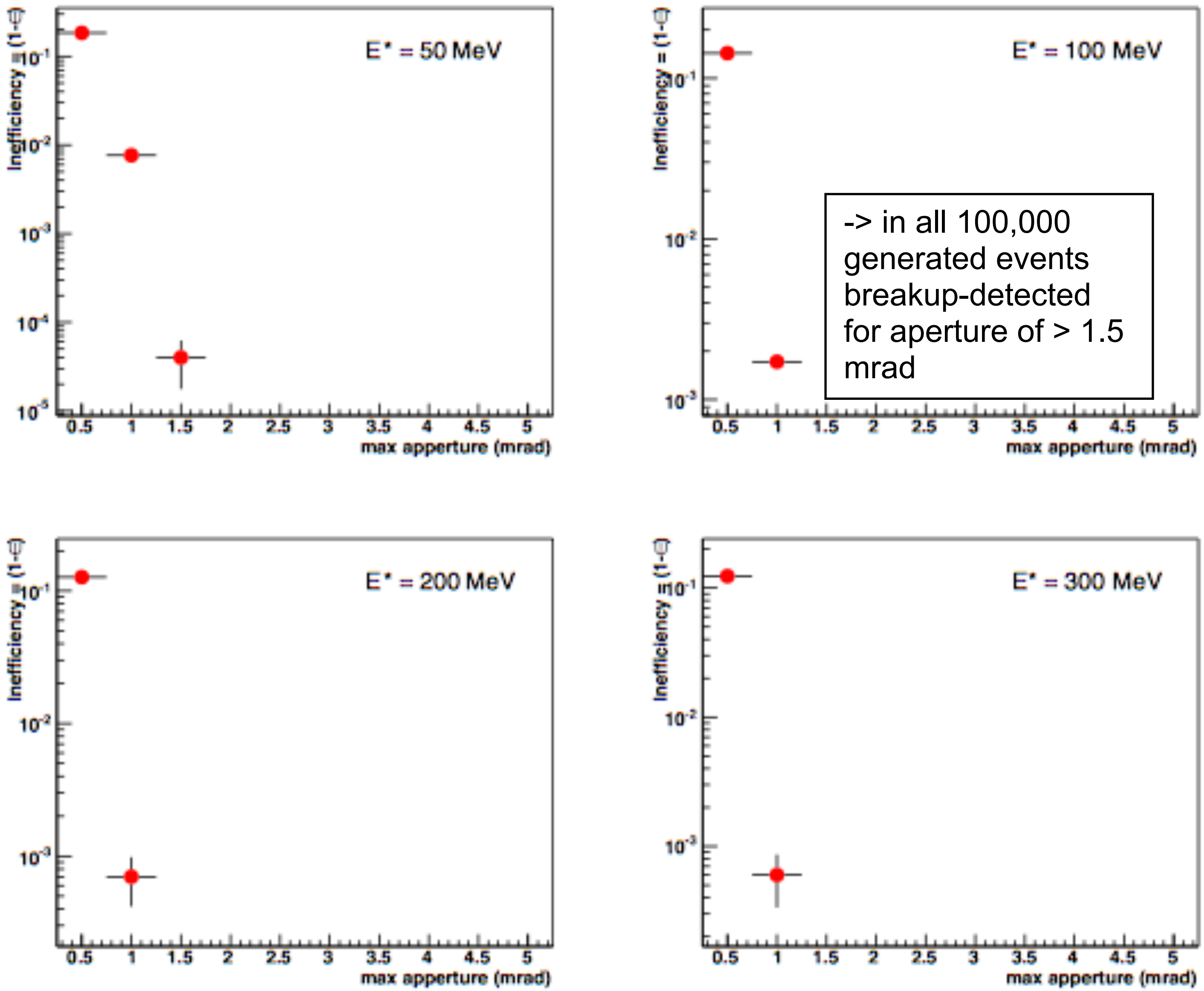
Plotted is not acceptance but inefficiency = (1 - fraction accepted)

Error bars are correct following TU and Z. Xu: arXiv:physics/0701199

Note: 100% detector eff. assumed

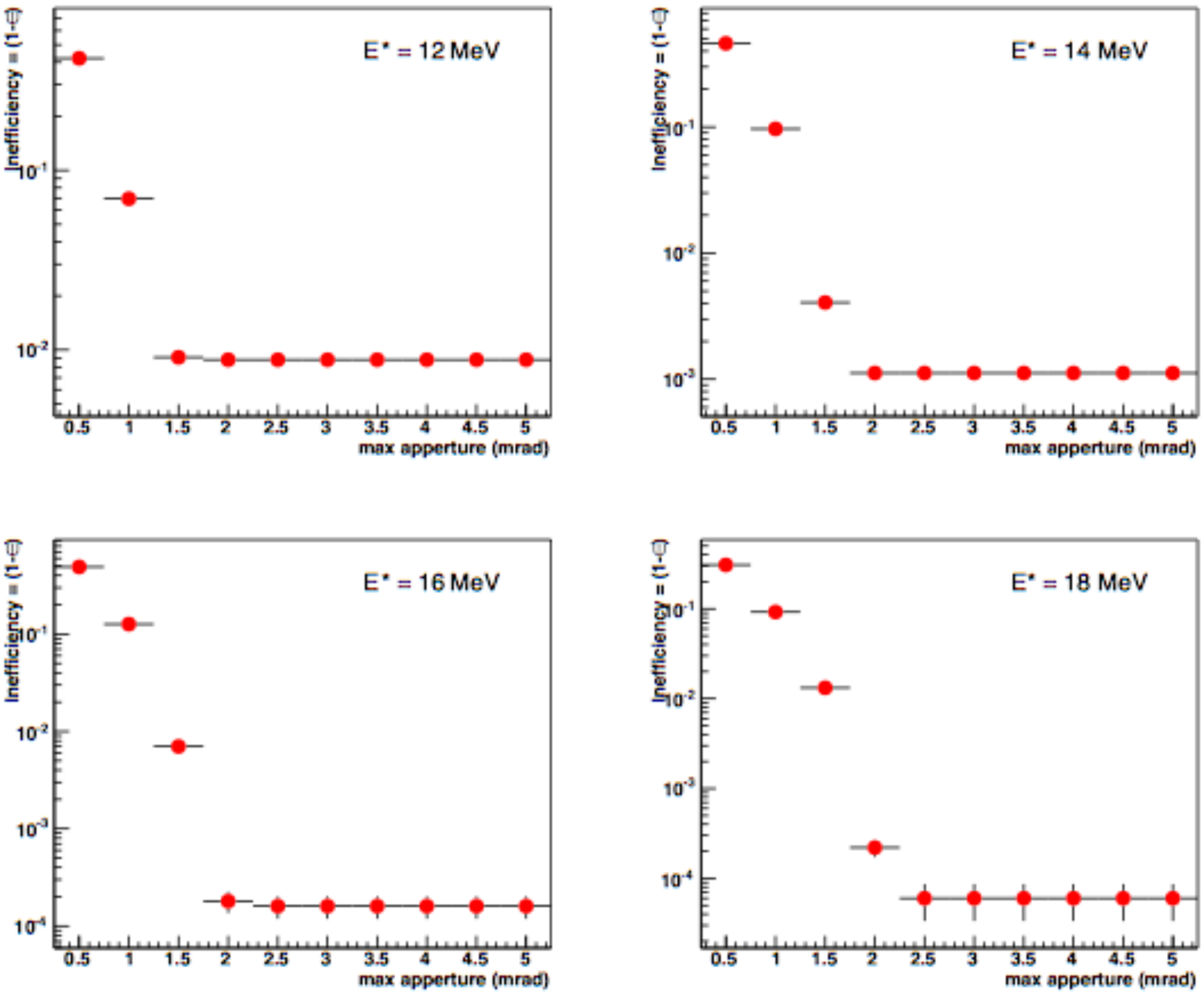
Acceptance studies

-> in all 100,000 generated events breakup-detected for aperture of > 1.5 mrad



Acceptance studies

Finer steps between 10 and 20 MeV



Factor 1000 (10^{-3}) reached at 2 mrad for E*=14 MeV

E* and how much rejection do we need?

According to Raju a good approximation is:

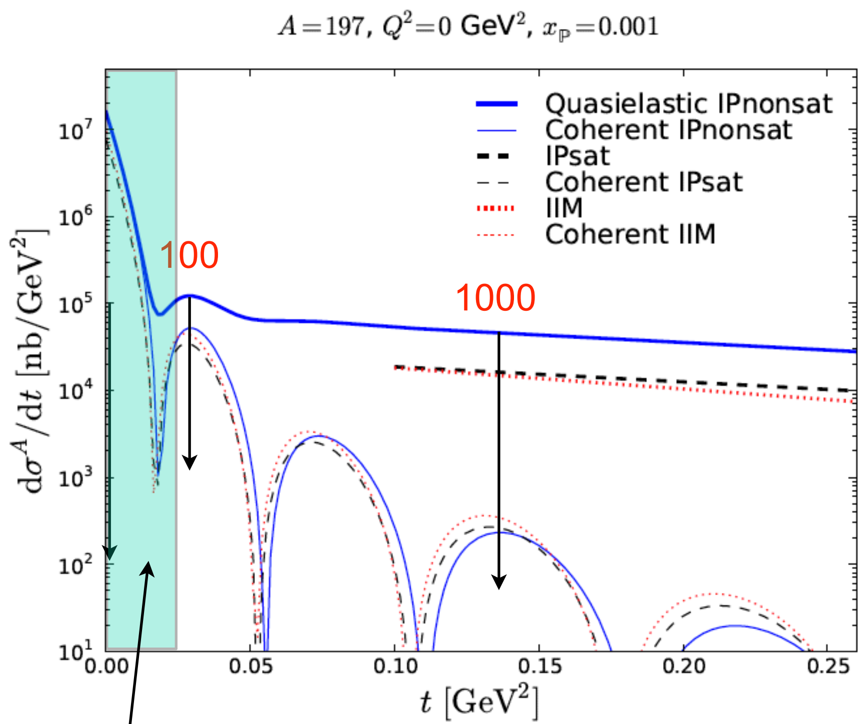
$E^* = -t / 2m_N$

T. Lappi '10

Rejection/Detection of breakup needed increases with *t*

Examples:

- t = 0.18 GeV^2 $\Rightarrow$ 95 MeV
  - can easily reach rejection of ~10^4
- 10 MeV $\Rightarrow$ 0.019 GeV^2
  - only factor 10 possible (with n)
- 12 MeV $\Rightarrow$ 0.023 GeV^2
  - factor 100
- 14 MeV $\Rightarrow$ 0.026 GeV^2
  - factor 1000



somewhat shaky range

Summary (for now)

- With an aperture of ±3 mrad we are in relative good shape even for 50 GeV Au beams
  - enough "detection" power for $t > 0.025$ GeV^2
  - below $t \sim 0.02$ GeV^2 we have to look into photon detection
    - Is it needed?
- All this assuming
  - Gemini++ is correct
  - $E^* \sim -t/2m_N$
  - I can make a ZDC 100% (>99.9999%) efficient
    - do we understand n detection on the 10^{-4} level?
- SMM and ABLA studies coming soon to verify Gemini++

Update

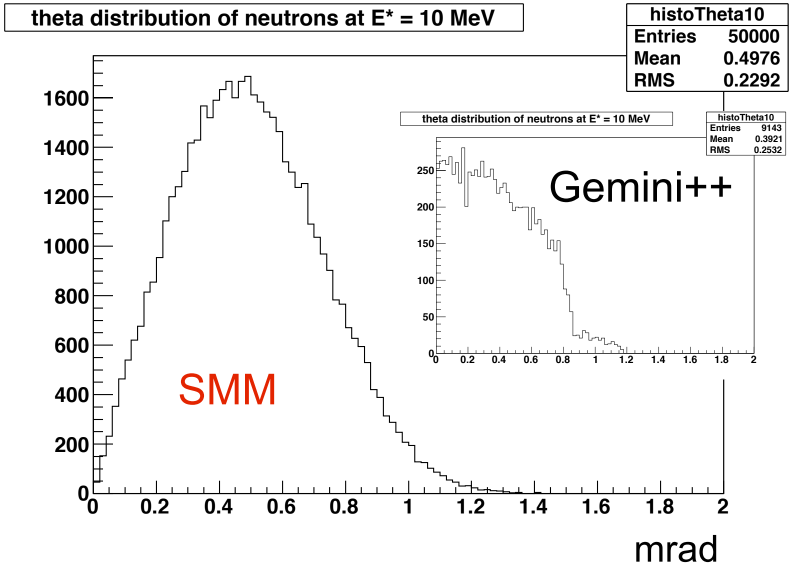
December 20, 2010

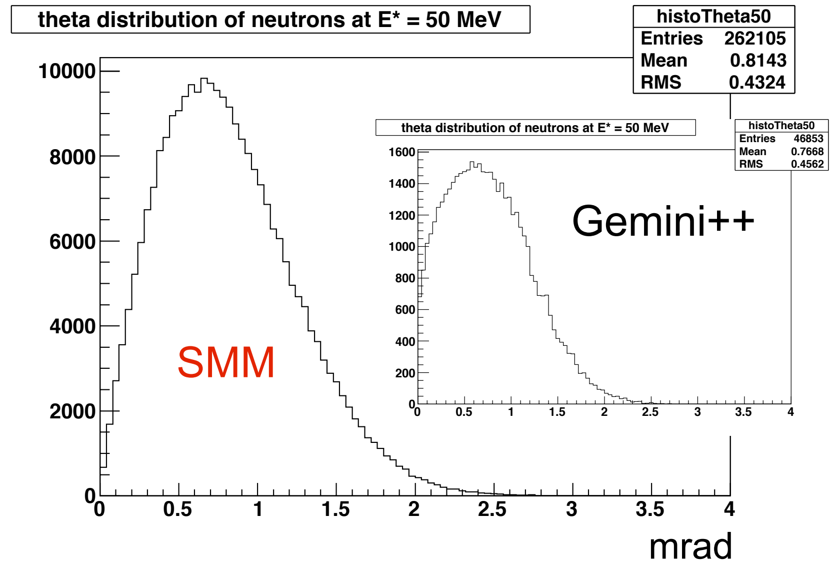
First look at SMM

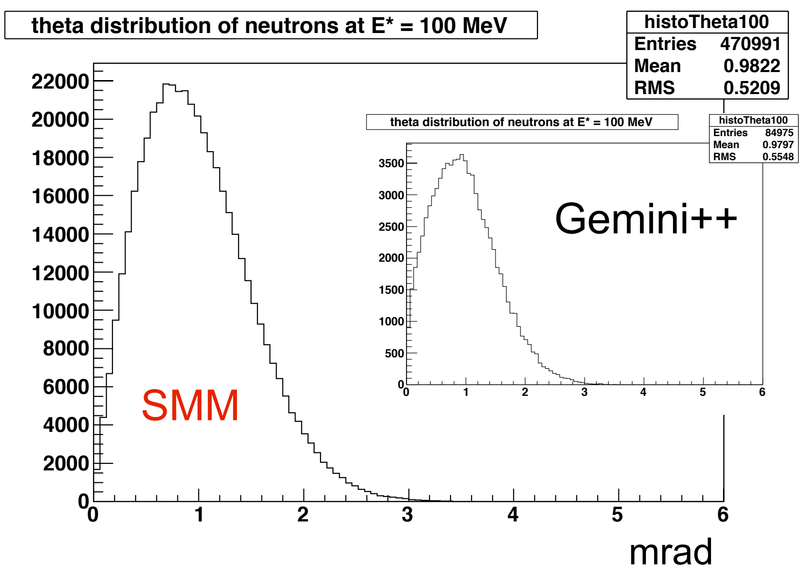
- 14k lines of unindented F77 and with max 5 char per variable
  - not easy to read/understand code
  - comments that ref. to changes to run on Soviet Vax ;-)
- Maintainer Alexander Botvina (GSI)
  - helpful, moderate responsive
- Code received from Davide Mancusi, University of Liège
  - Davide added modifications that makes running SMM easier and a bit closer to Gemini++
  - very helpful but not an expert on SMM
- Calling SMM from C++ main program
  - can use ROOT and make same plots as for Gemini++

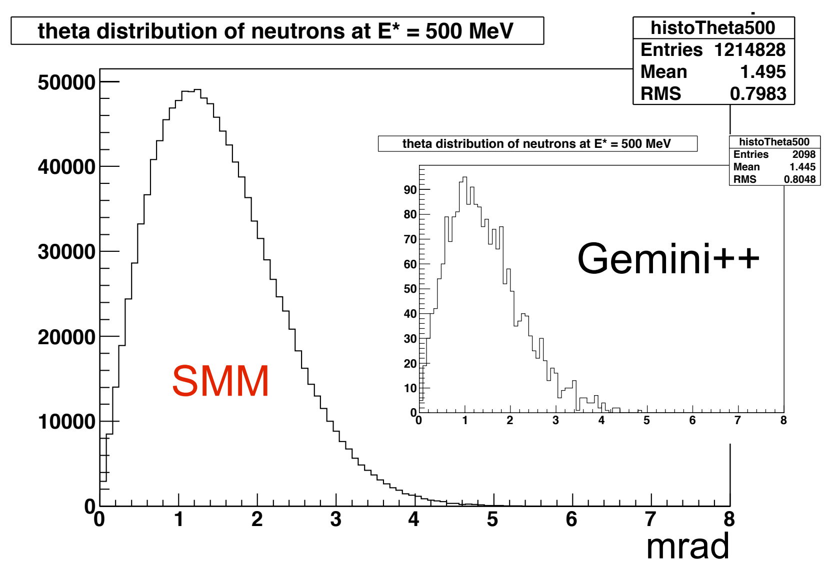
First Look at SMM

50 GeV Au beam

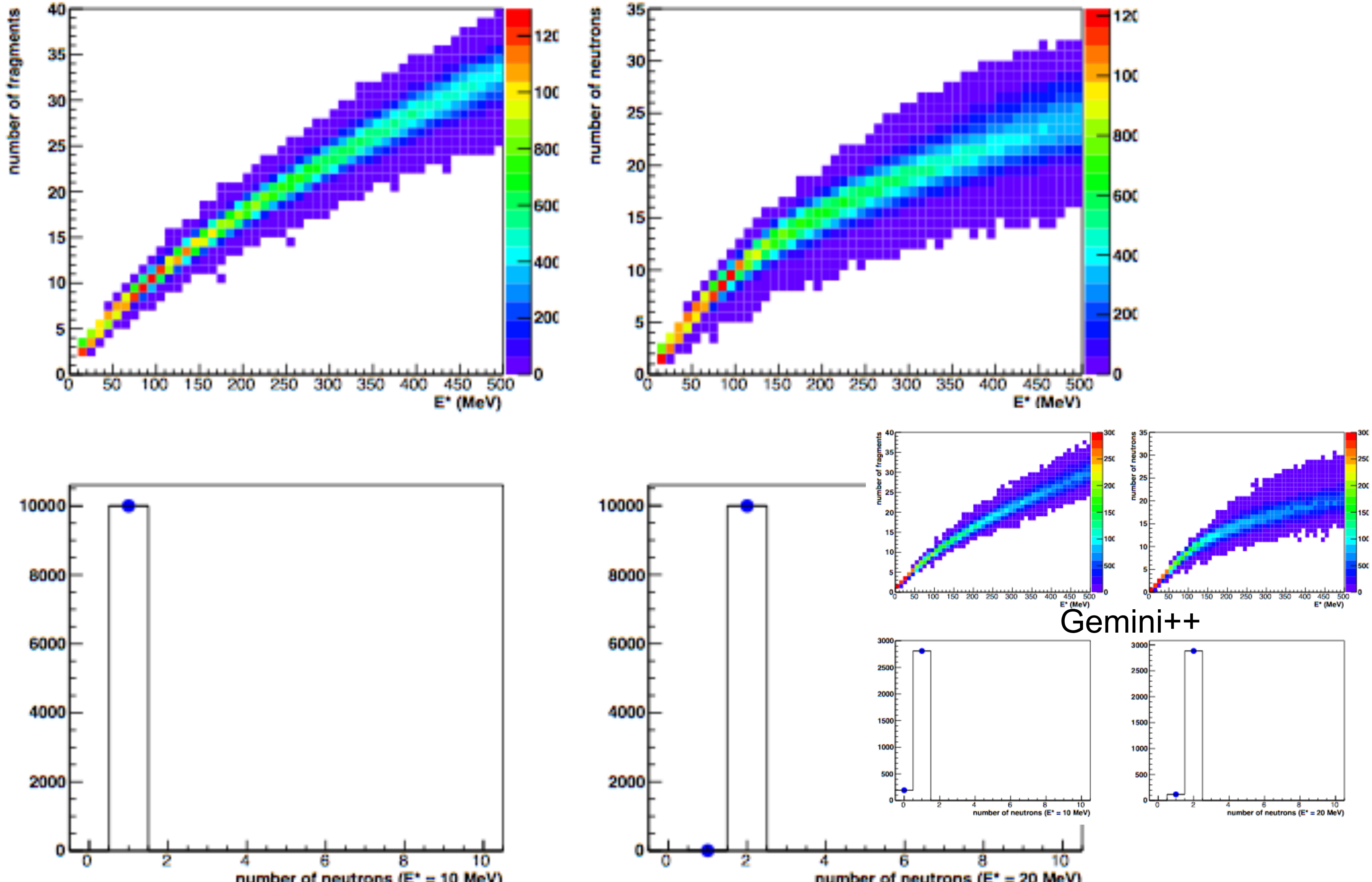




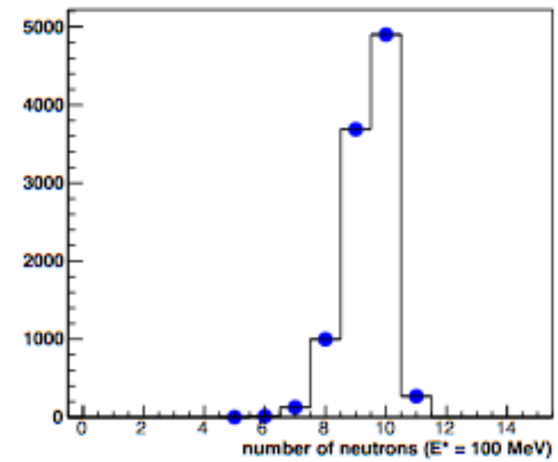


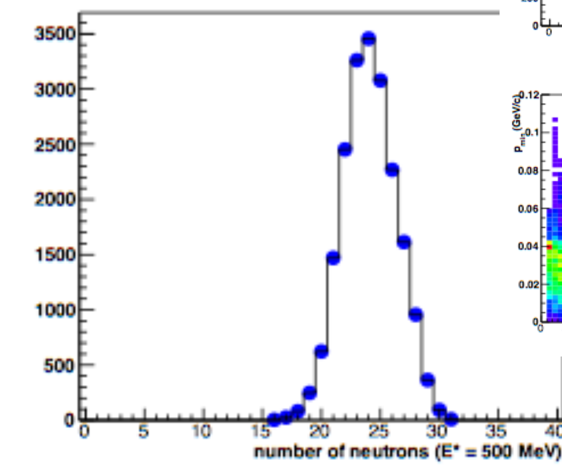


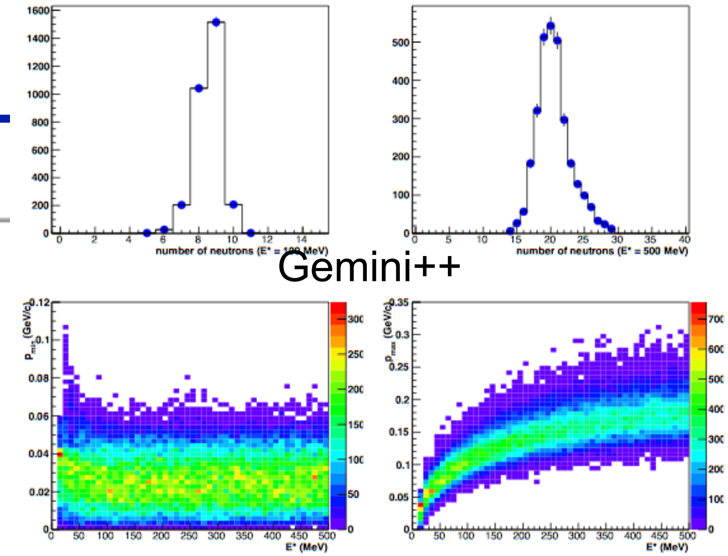
SMM

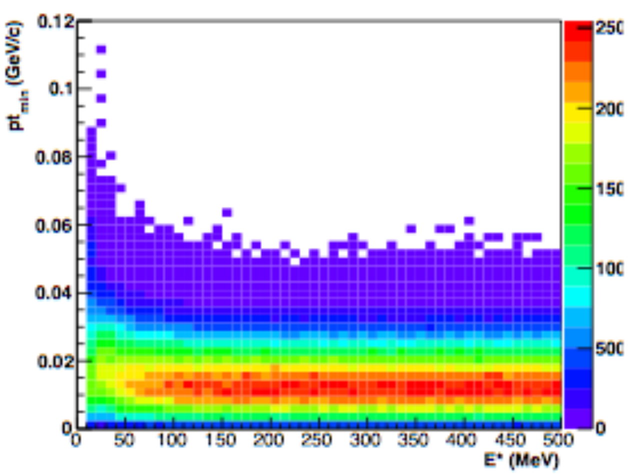

number of fragments
E* (MeV)
number of neutrons
E* (MeV)
number of neutrons (E* = 10 MeV)
number of neutrons (E* = 20 MeV)
Gemini++

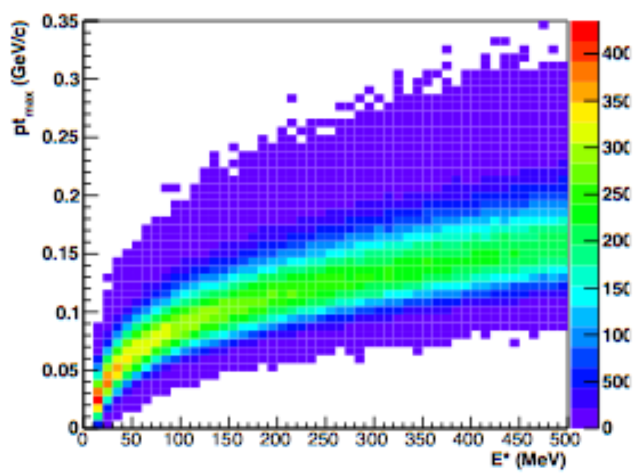
SMM







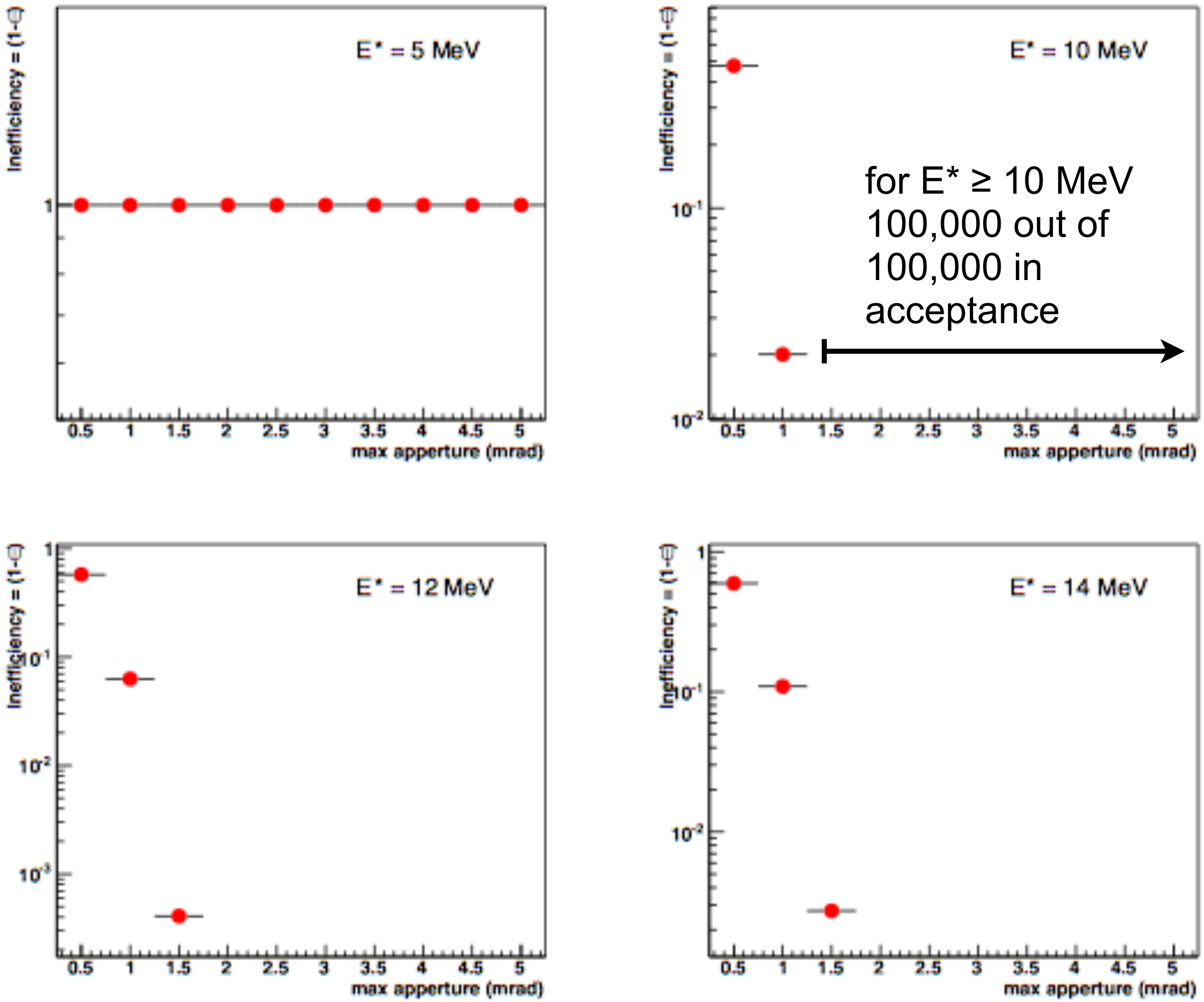




SMM and Gemini++ give very similar results

Acceptance studies using SMM

50 GeV Au beams

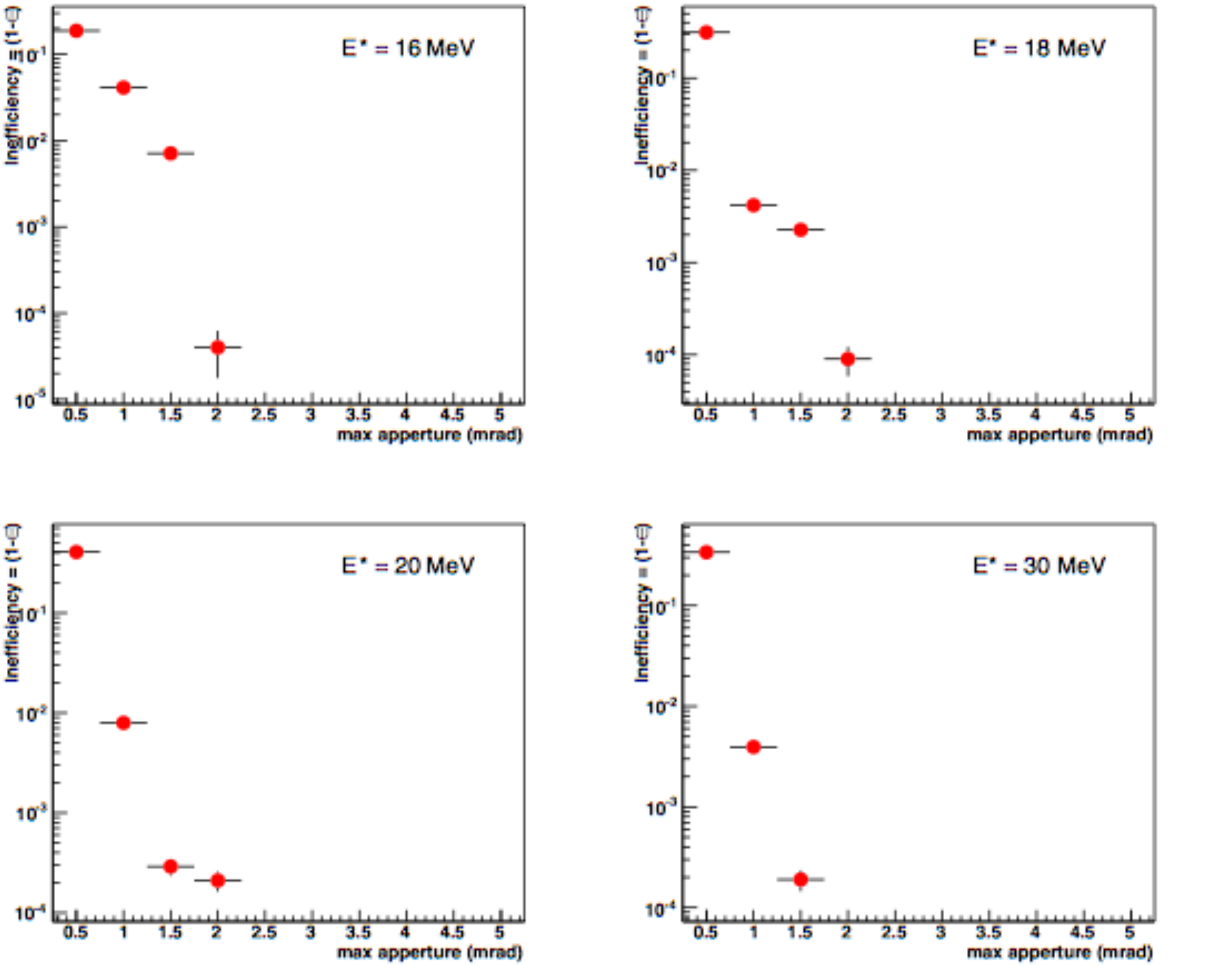


Plotted is not acceptance but inefficiency = (1 - fraction accepted)

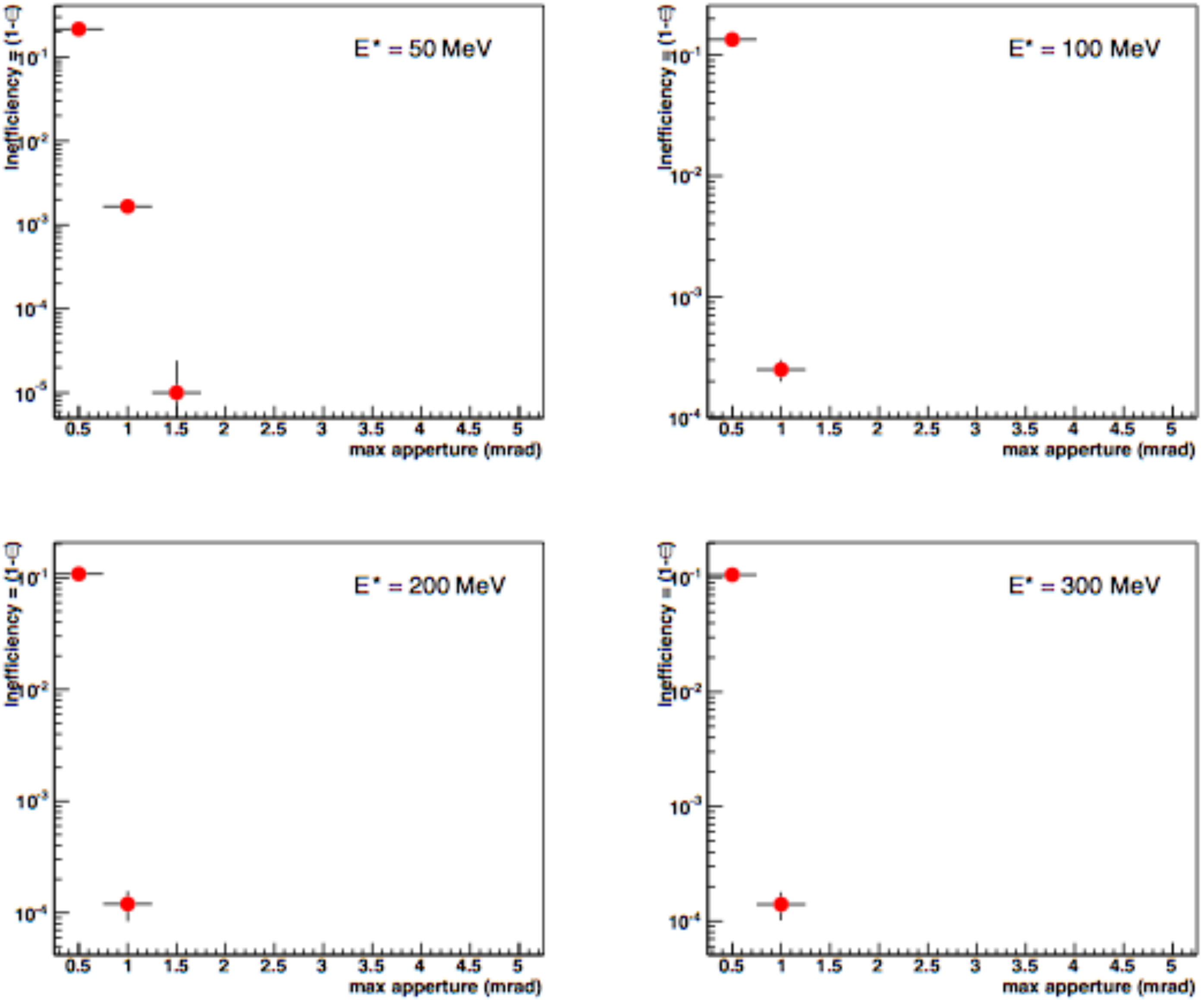
Note: 100% detector eff. assumed

100,000 events generated for each E*

Acceptance studies using SMM



Acceptance studies using SMM



Summary SMM:

SMM confirms GEMINI++ results

- SMM even more optimistic
- For $E^* \geq 10$ MeV and 2.5 mrad acceptance SMM indicates we have rejection power of at least 10^5.
- SMM (as Gemini++) produces no photons
  - it's however more optimistic than Gemini++ at $E^* = 10$ MeV

Excitation Energy, E*, and t

Raju:

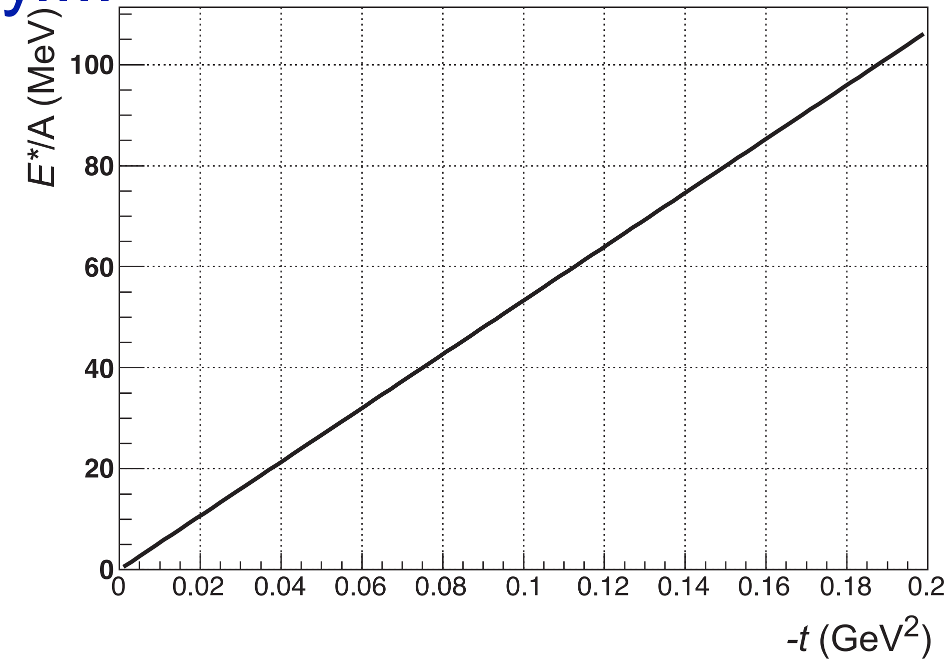
$E^*_A = E^*_{tot} = -t_A / (2\ m_p{*}A)$

$= -A^2\ t / (2\ m_p{*}A) = -A\ t / (2\ m_p)$

essentially:

$E^*/A = -t / 2m_N$

making life super easy....

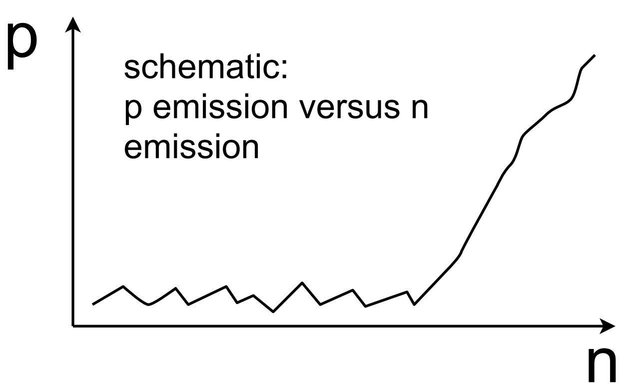


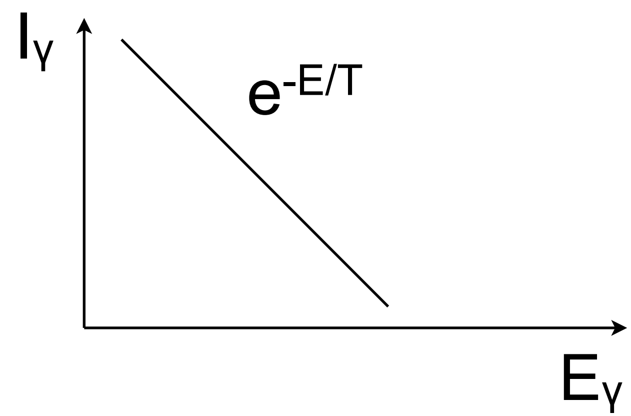
Discussion with Wolfgang Trautman (GSI)

Use Gemini for E*/n < 3 MeV (NB: Gemini++ might have overcome issues)

use SMM for E*/n > 3 MeV

Excitation time: 100 fm/c per 5 MeV/n





EAL (evaporation attractor line)
E*/A > 3 MeV
20 fm/c multifragmentation

Stat. γ radiation

γ emission up to max 6 MeV then n-emission

Update

December, 2015

How Sartre deals with E*

Incoherent Diffraction:

Generate hadron dissociation mass according to

$$dN/dM_Y^2 \sim 1/M_Y^2$$

as done in HERA simulations (H. Jung, P. Newman)

- Lower bound is the invariant mass of the hadron beam particle
- upper bound is some arbitrary (big) value

Our model of incoherence is that the difference of the diffractive mass of one (1) proton out of the nucleus gives the final excitation energy E*.

Technical Note: We calculate and quote eA kinematics always in units of 'per nucleon'.. Hence we have to calculate E* and divide it by A to keep the kinematic consistent.

How Sartre deals with E* - Step 1

In `ExclusiveFinalStateGenerator::generate()`

```
if (mIsIncoherent && mA > 1) {
     const double lower = hMass2;
     const double upper = 9; // GeV2
     mMY2 = lower*upper/(upper - rndm->Uniform()*(upper-lower));
     double MY_per_nucleon = (sqrt(hMass2)*(mA-1) + sqrt(mMY2))/mA;
     mMY2 = MY_per_nucleon*MY_per_nucleon;
     if (mMY2 < hMass2) mMY2 = hMass2;
 }
 else {
     mMY2 = hMass2;
 }
```

Now we have M_Y^2

Note the cheat: the actually M_Y is divided by A although it come from only one proton.

How Sartre deals with E* - Step 2

With the outgoing “virtual” proton having a mass > m_p we need to adjust the whole kinematic. Cannot just invent mass and keep kinematic intact. There’s no analytic solution. Need to run a root finder that does not need derivates but uses a bracketing algorithm (Brent).

Input: t, m_V, m_Y, φ, $\mathbf{p}_{in}$, $\mathbf{p}_{\gamma^*}$

Output: new scattered proton energy

In `ExclusiveFinalStateGenerator::generate()`

```
//
//  Outgoing proton (hadron) system
//
pz = (mT- hMass2 - mMY2 + 2*mHadronBeam.E()*E)/(2*mHadronBeam.Pz());
pt = sqrt(E*E-pz*pz-mMY2);
px = pt*cos(phi);
py = pt*sin(phi);
TLorentzVector theScatteredProton(px, py, pz, E);
```

Now we have $\mathbf{p}_{out}$

How Sartre deals with E* - Step 3

Now the breakup code (here shortened) does:

$$E^* = (\text{invariantMass}(\mathbf{p}_{out}) - m_p) \times A$$

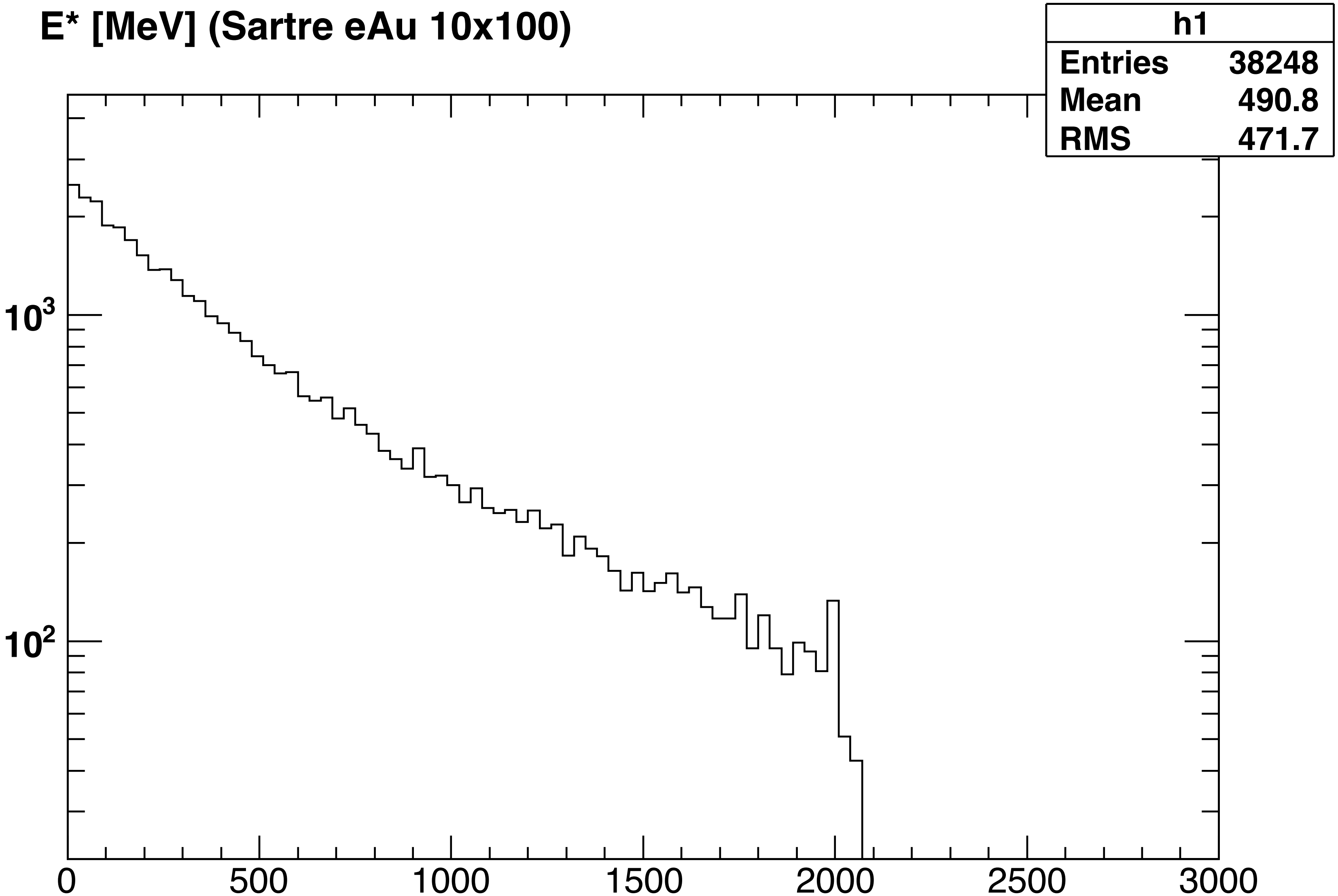
(times A since we were in the “per/nucleon” mode, that’s a cheat - we are not using A times the diffractive mass/energy)

```
int FrangibleNucleus::breakup(const TLorentzVector& dissSystem)
{
    EventGeneratorSettings *settings = EventGeneratorSettings::instance();

    //
    //  Estimate excitation energy
    //  Note that the dissSystem is given in units of 'per nucleon'.
    //  Hence we have to multiply the excitation energy with A.
    //
    mExcitationEnergy = (dissSystem.M()-protonMass)*mA;
    double Ex = mExcitationEnergy;

    Ex *= 1000;  // Gemini uses the total energy in MeV
```

Now we have E* and use that in Gemini++

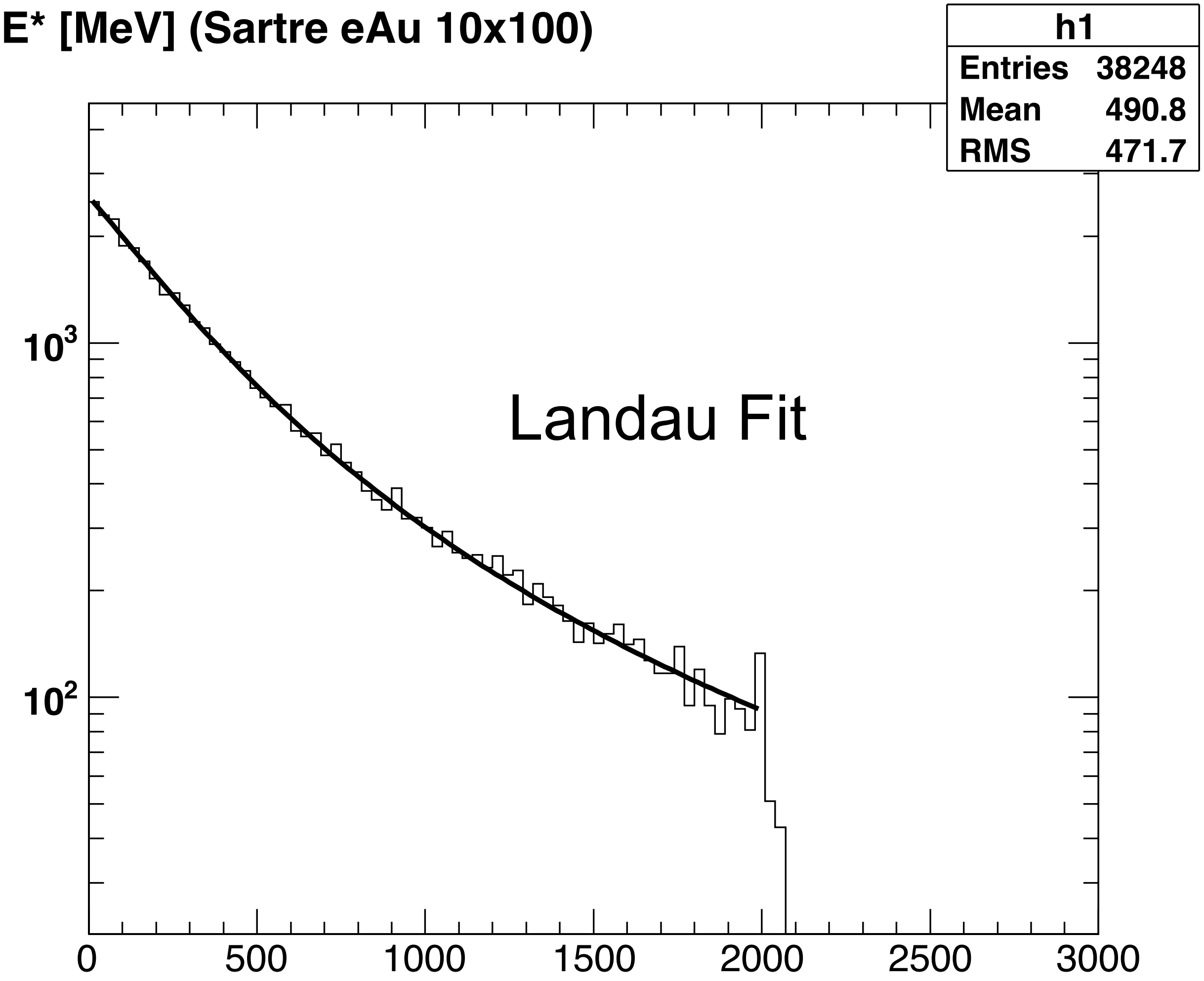
How Sartre deals with E* - Summary

- We are **not** following Raju's estimate of E*/A = -t / 2mN
- We follow HERA in that we independently of the process create a diffractive mass MY and then adjust the kinematic of the process accordingly. The only final state particles affected are:
  - outgoing "scattered" hadron
  - vector meson
- We assume that the diffractive energy generated by a single proton is "somehow" distributed in the system and used to evaporate the nucleus



Sartre 100k events

E* [MeV] (Sartre eAu 10x100)




```
root [22] TF1 *fa1 = new TF1("fa1","[0]*TMath::Landau(x,[1],[2])",0,2000);
root [23] fa1->SetParameters(2.04676e+04, -1.94841e+02, 1.33551e+02);
root [24] fa1->Draw("same");
```

100 GeV Au - Gemini++ & Emittance

Values taken from eRHIC Design Study Document

- rms normalized emmittance $\varepsilon_N = 0.2\ 10^{-6}$ m, β^*=5cm
  - some confusion about factor π that got solved (see below)
- $\theta = [\varepsilon_N / (\gamma\beta^*)]^{1/2}$ = 0.2 mrad (0.283 mrad for 50 GeV/c)
- Gemini++ works in nucleus rest frame
  - boost by beam vector with a gaussian spread of σ_θ = 0.2 mrad

Hi, Thomas,

your calculation or rms angle spread (theta) is absolutely correct.

Sorry, for confusion, but from some time ago we've started to use the emittance unit without mentioning the factor pi. This unit is standard for all electron accelerators as well as for many proton machines, including LHC.

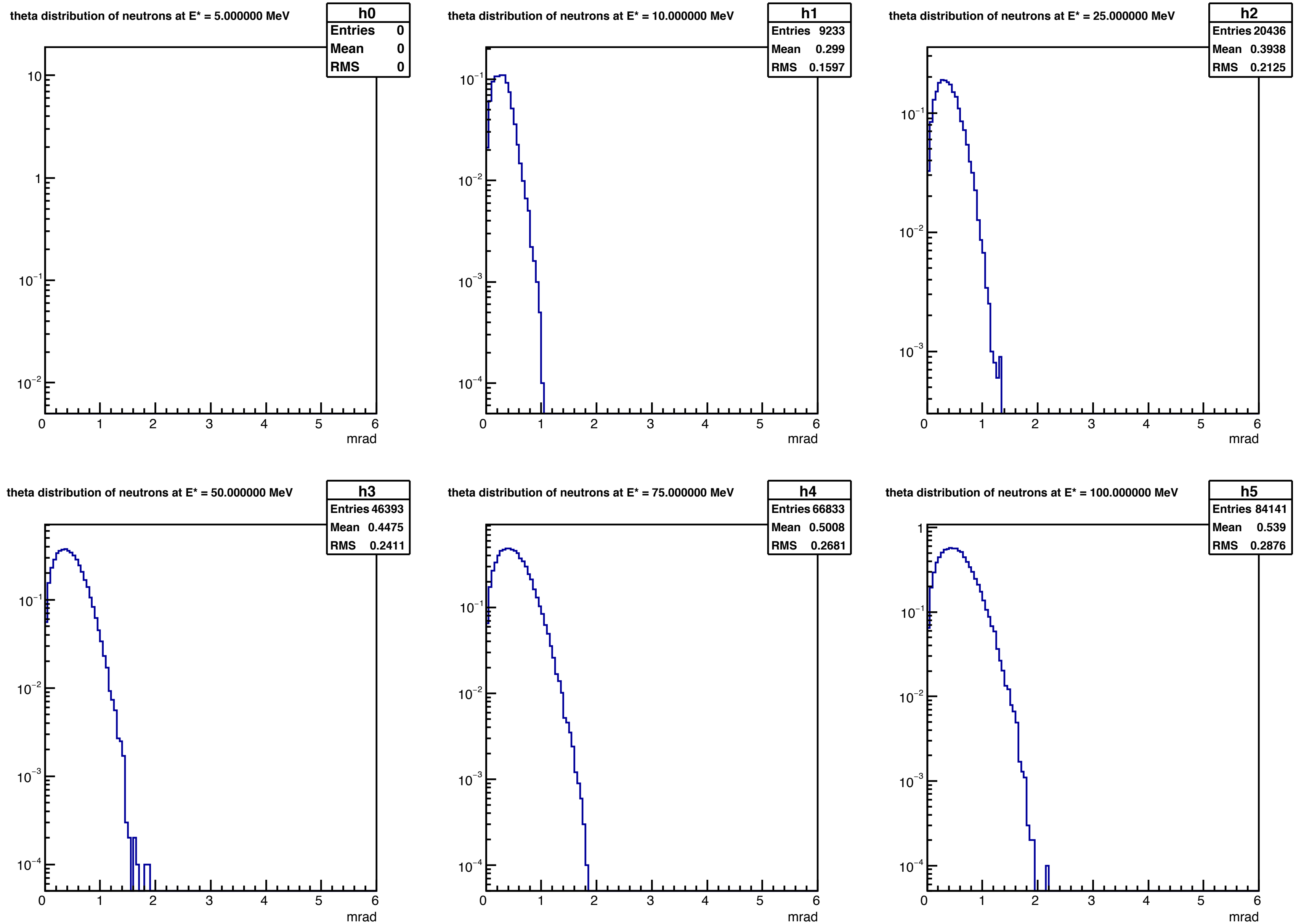
So, if you see (and will see) in the present eRHIC parameters tables the emittance 0.2 10^-6 m , it means absolutely the same emittance of 0.2 10^-6 pi m from an old table, the factor pi is just not mentioned.

With best regards,

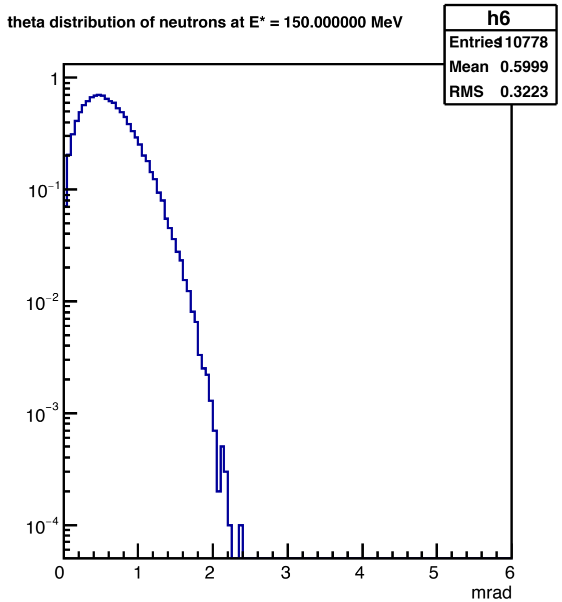
Dr. Vadim Ptitsyn

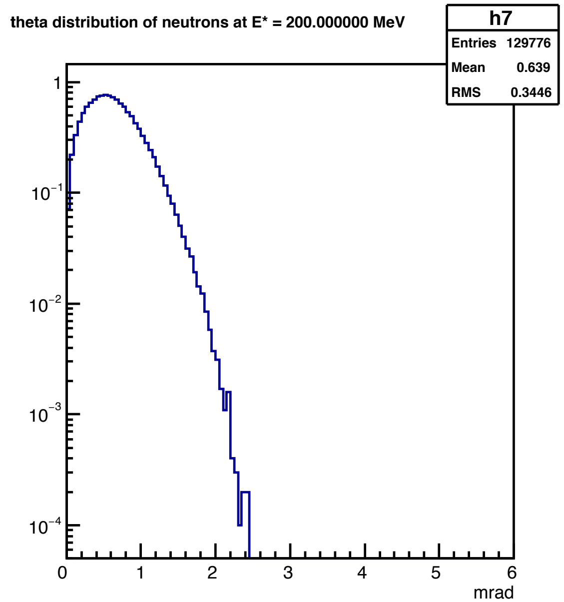
Au 100 GeV

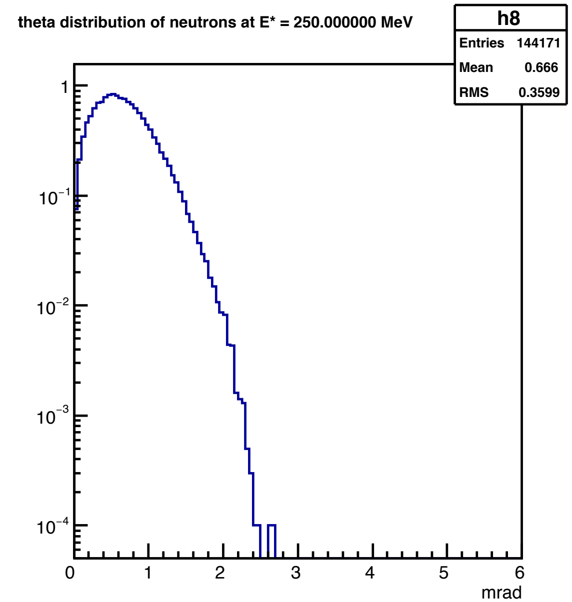
Results: θ Distribution of Neutrons (I)

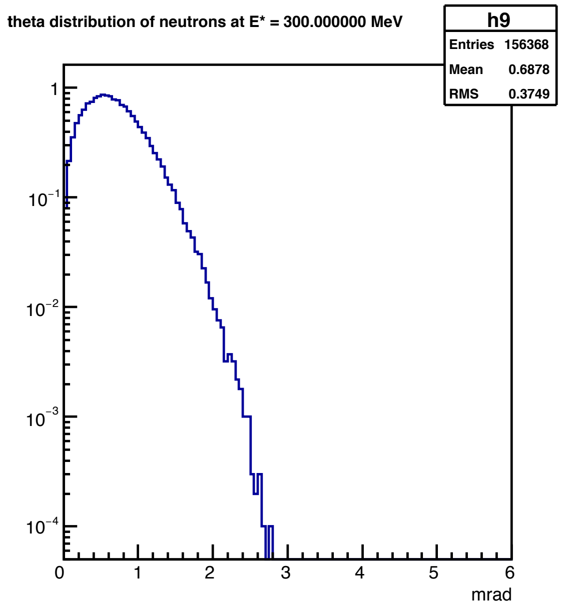


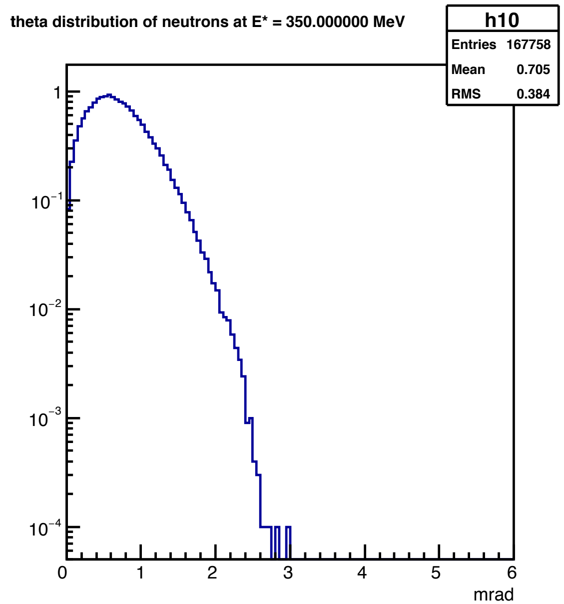
Results: θ Distribution of Neutrons (II)

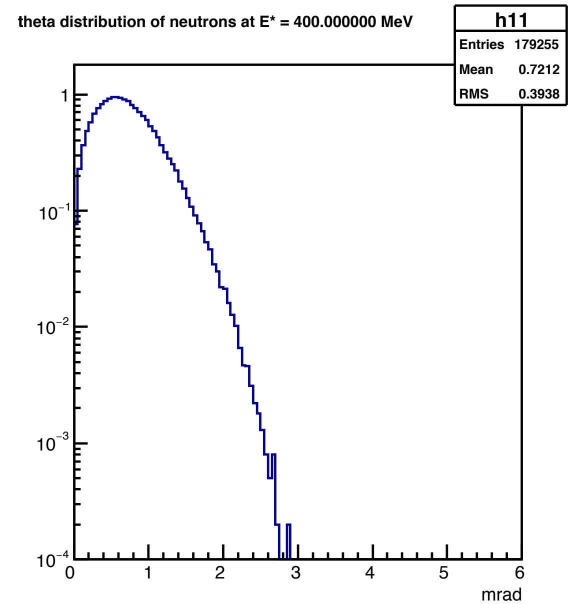




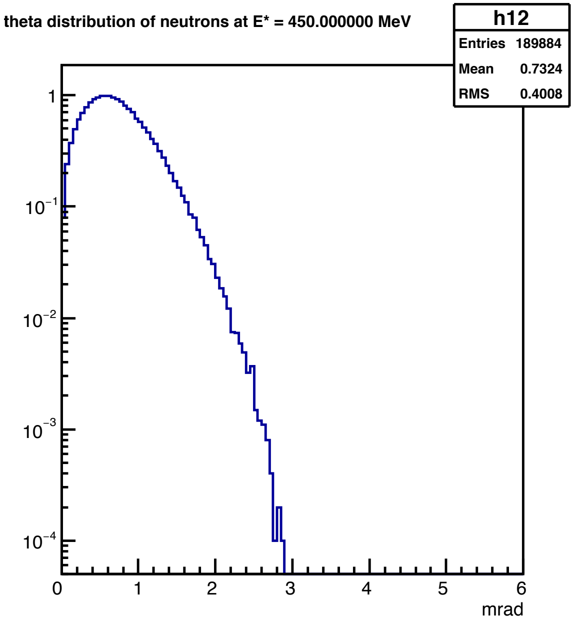


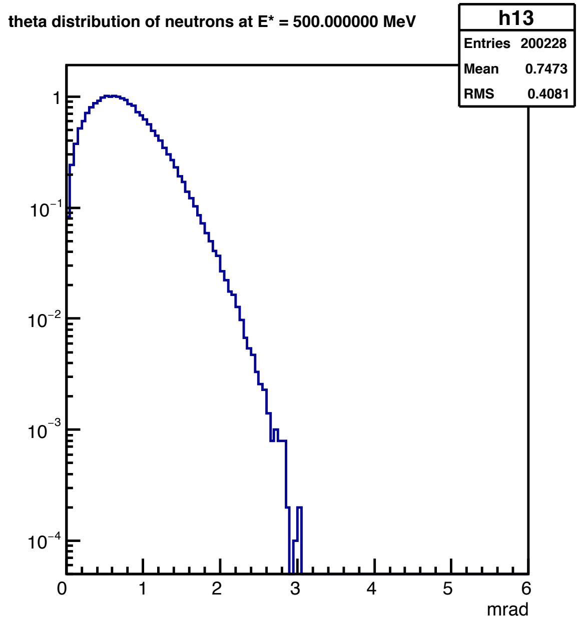


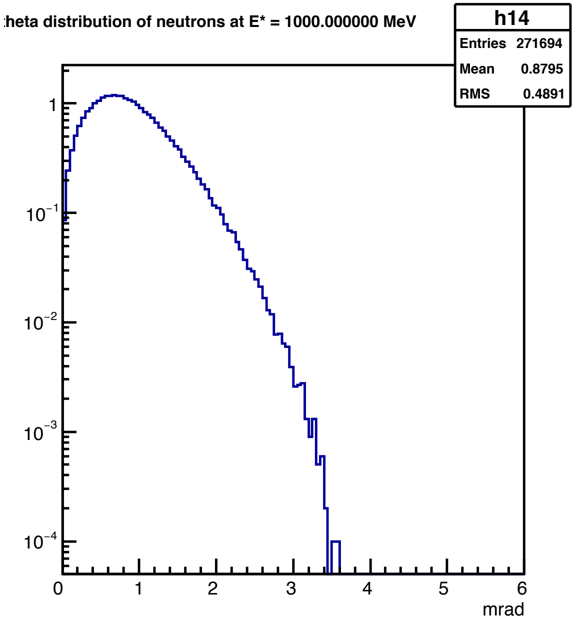




Results: θ Distribution of Neutrons (III)

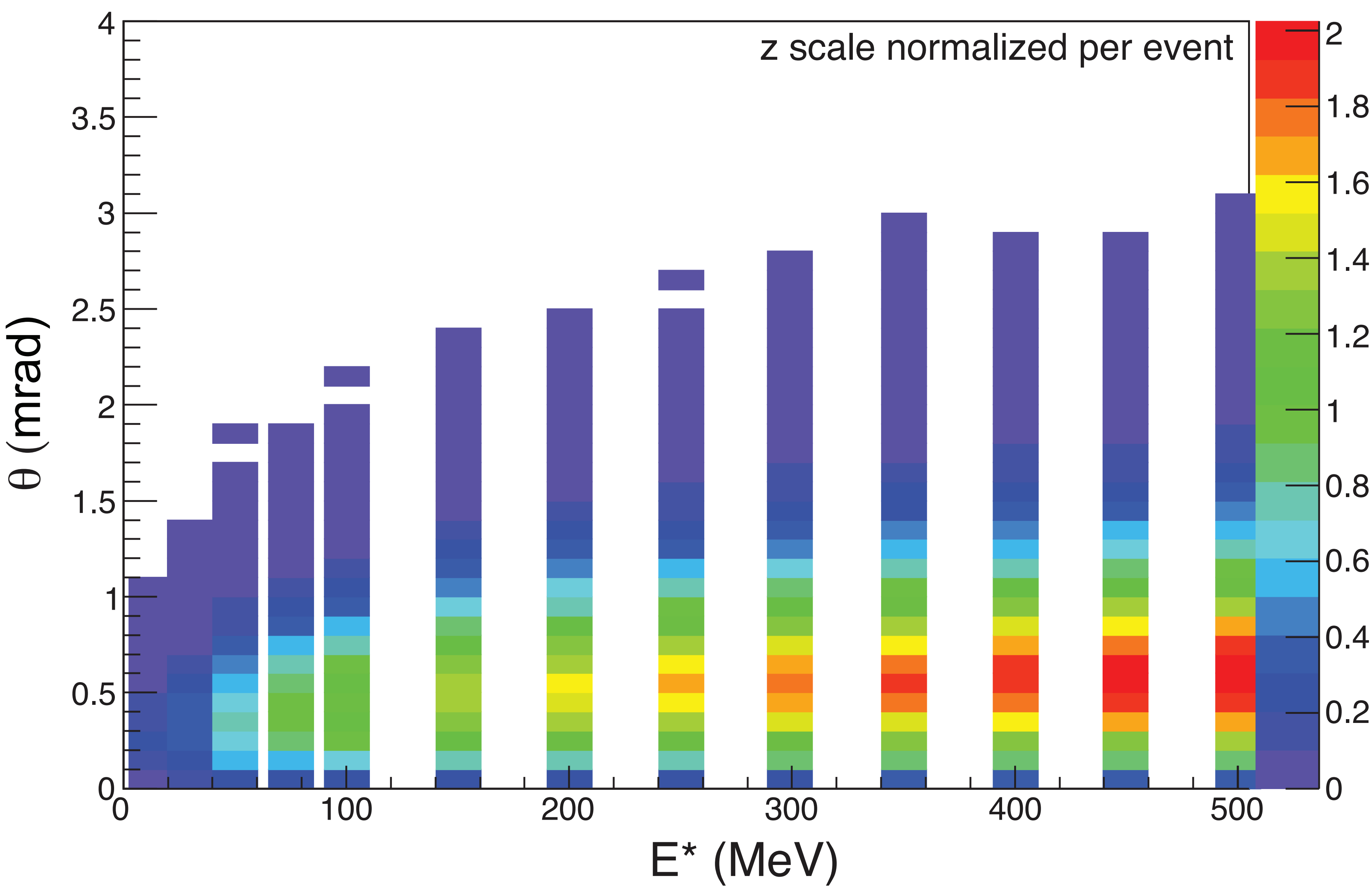




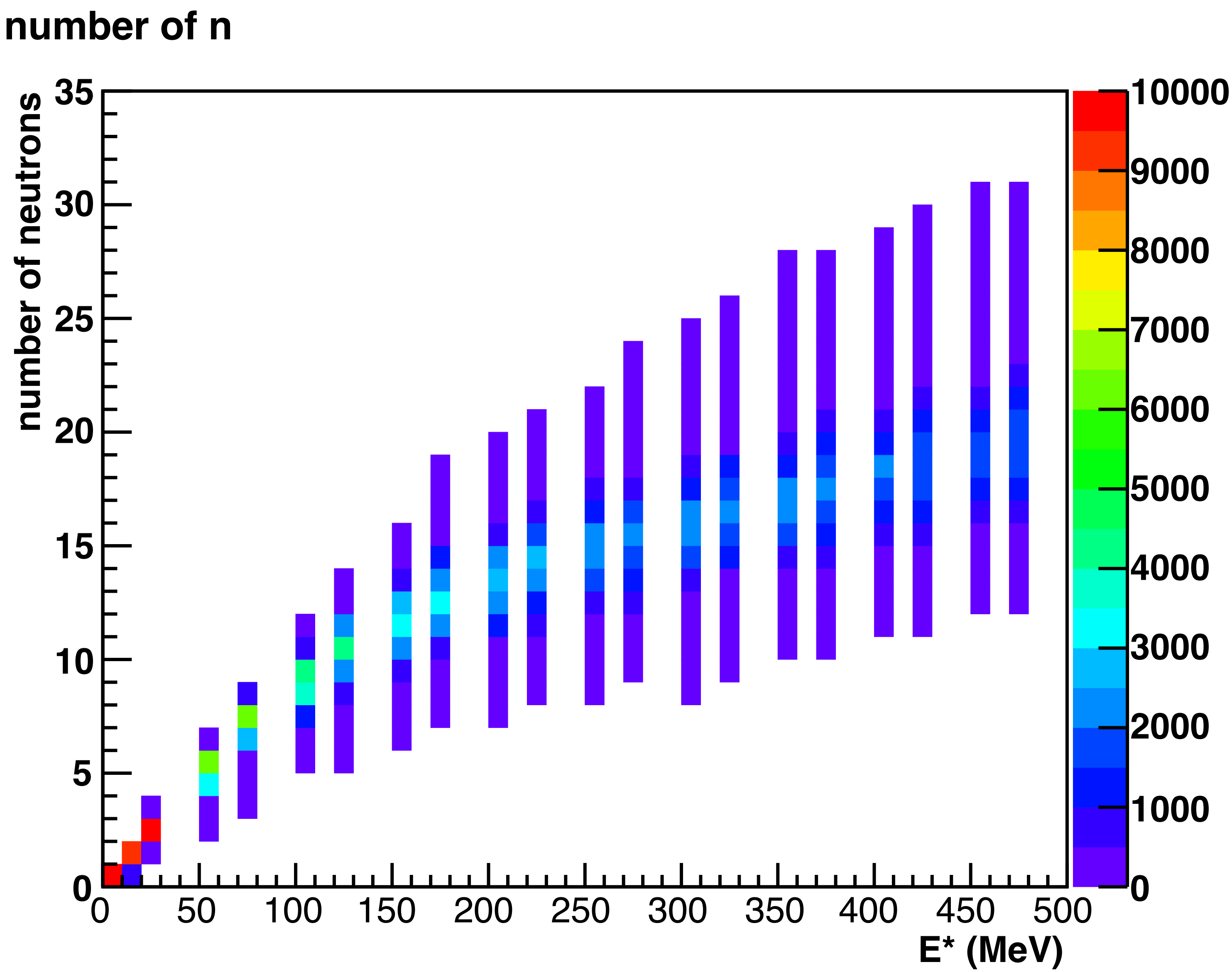


Note: above E* = 100 MeV the Median (max) is not shifting.

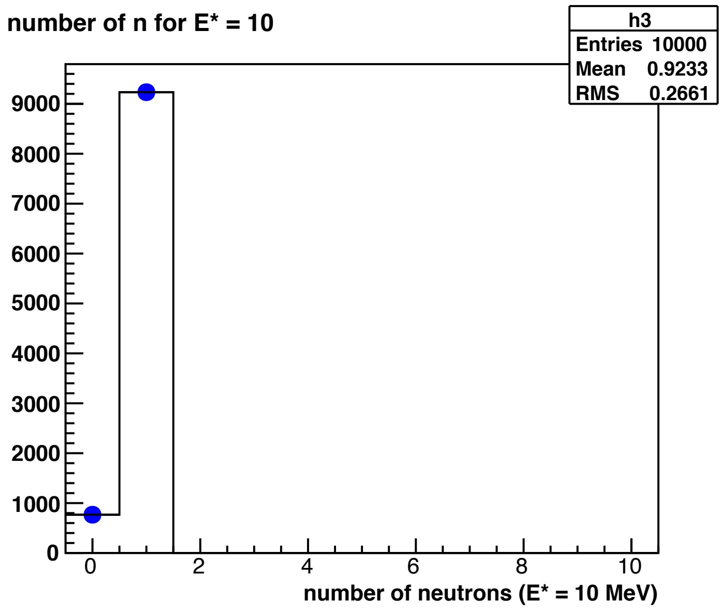
Results: θ vs E* (⟨per event distribution⟩)

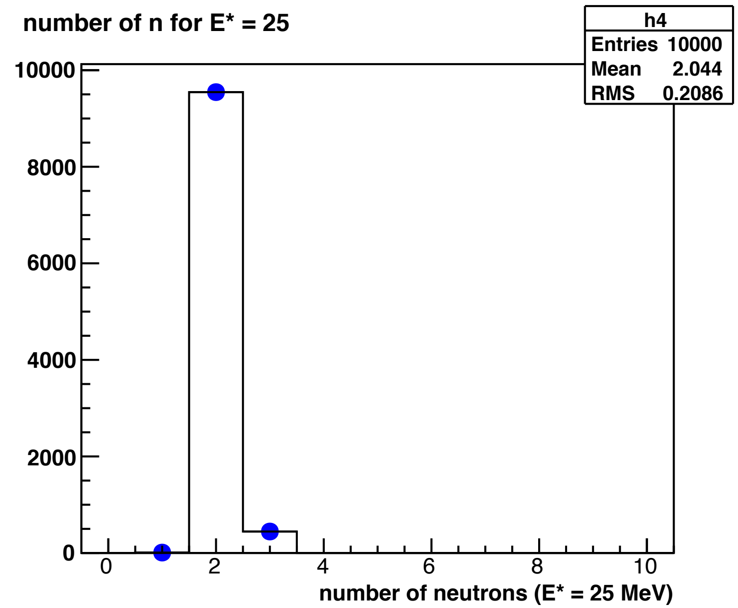


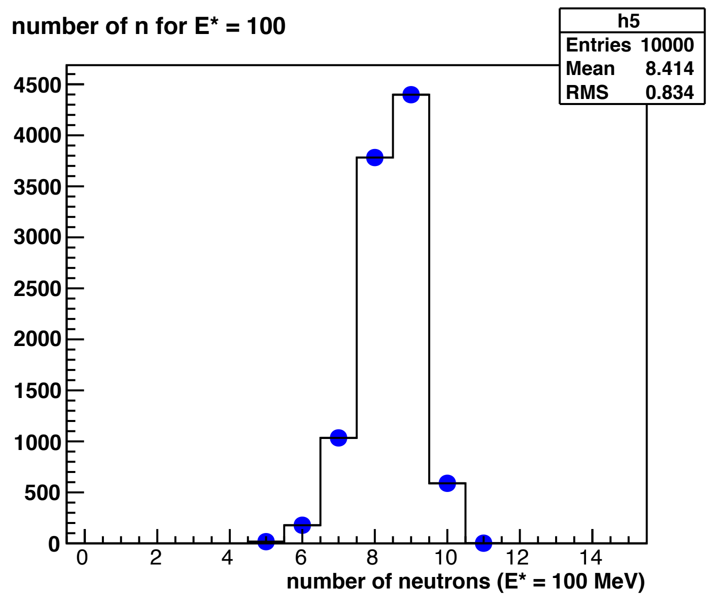
Results: Number of Neutrons (I)

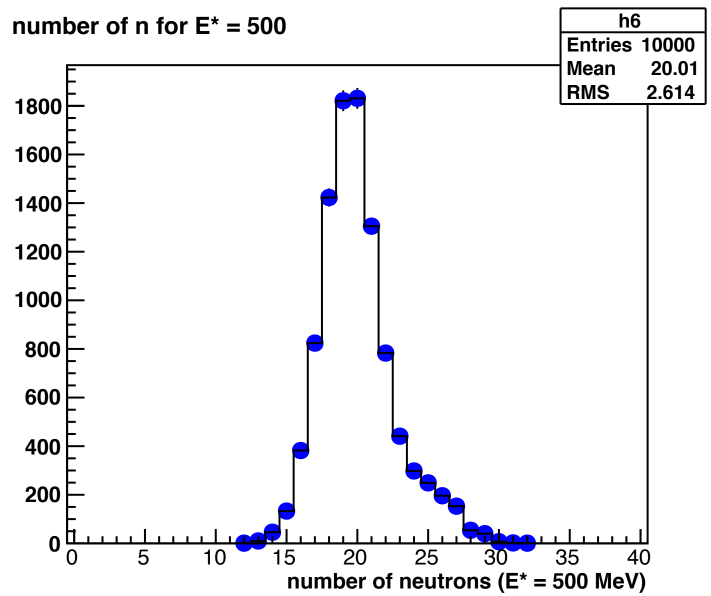


Results: Number of Neutrons (II)

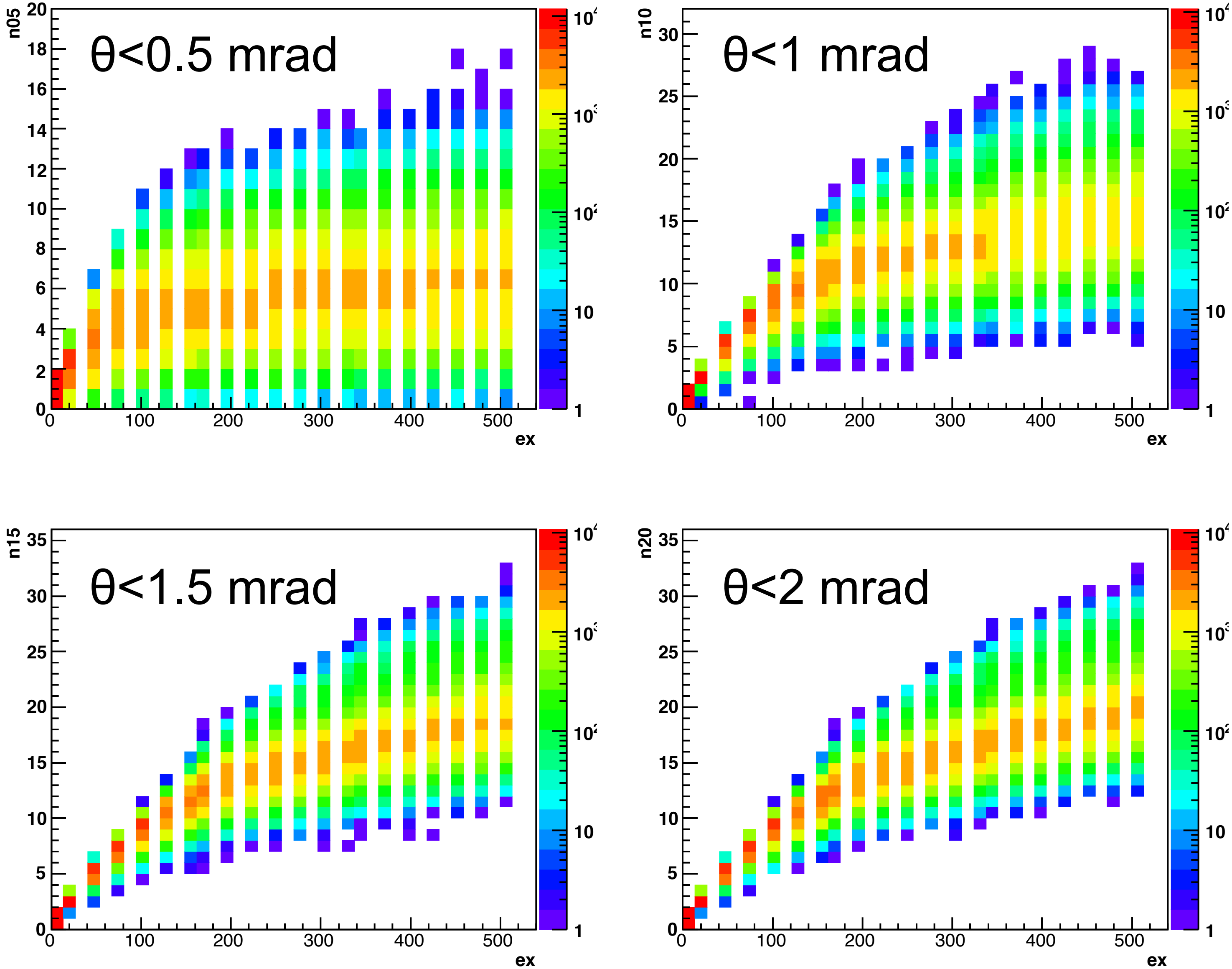




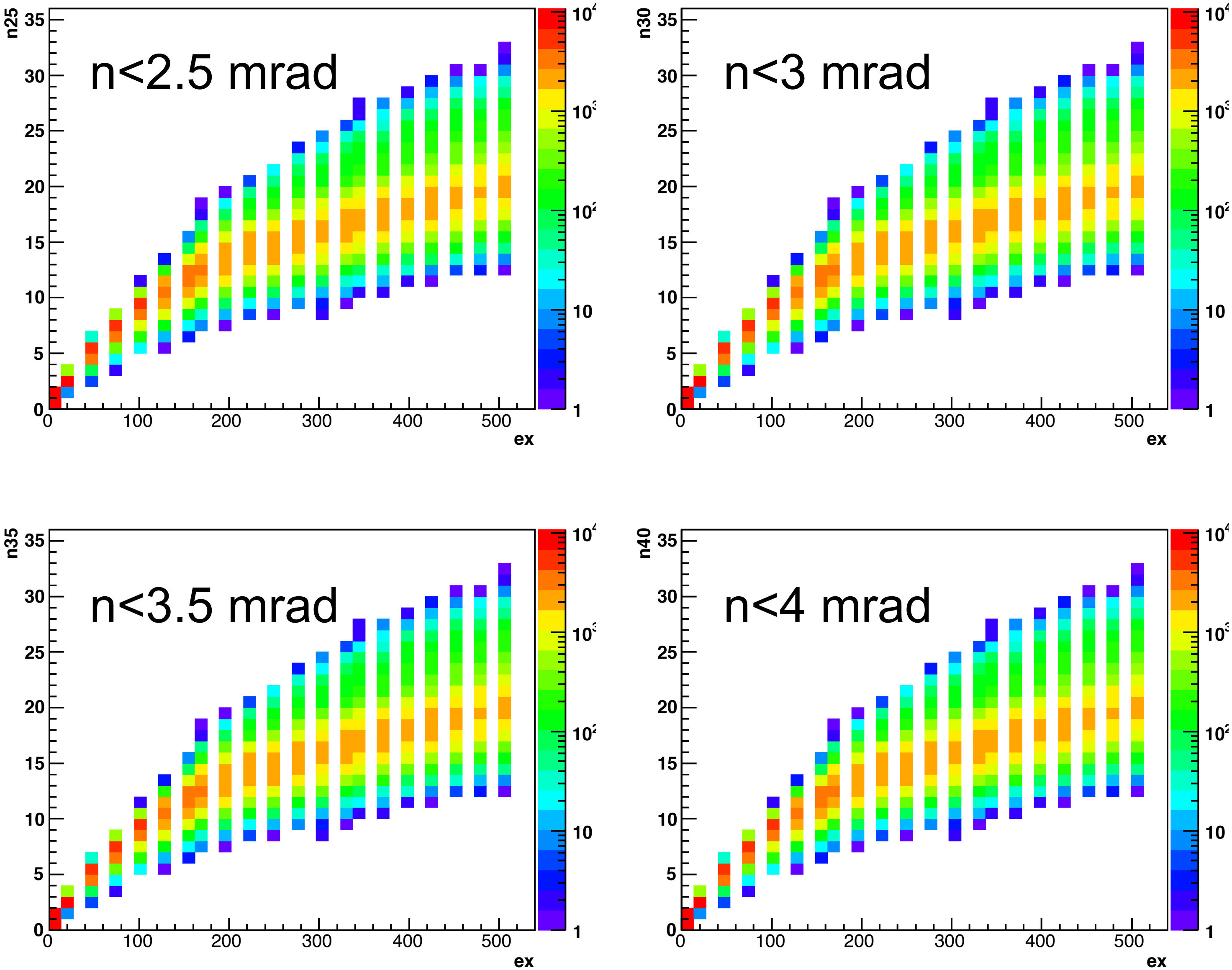




Results: # of Neutrons below given Δθ (I)



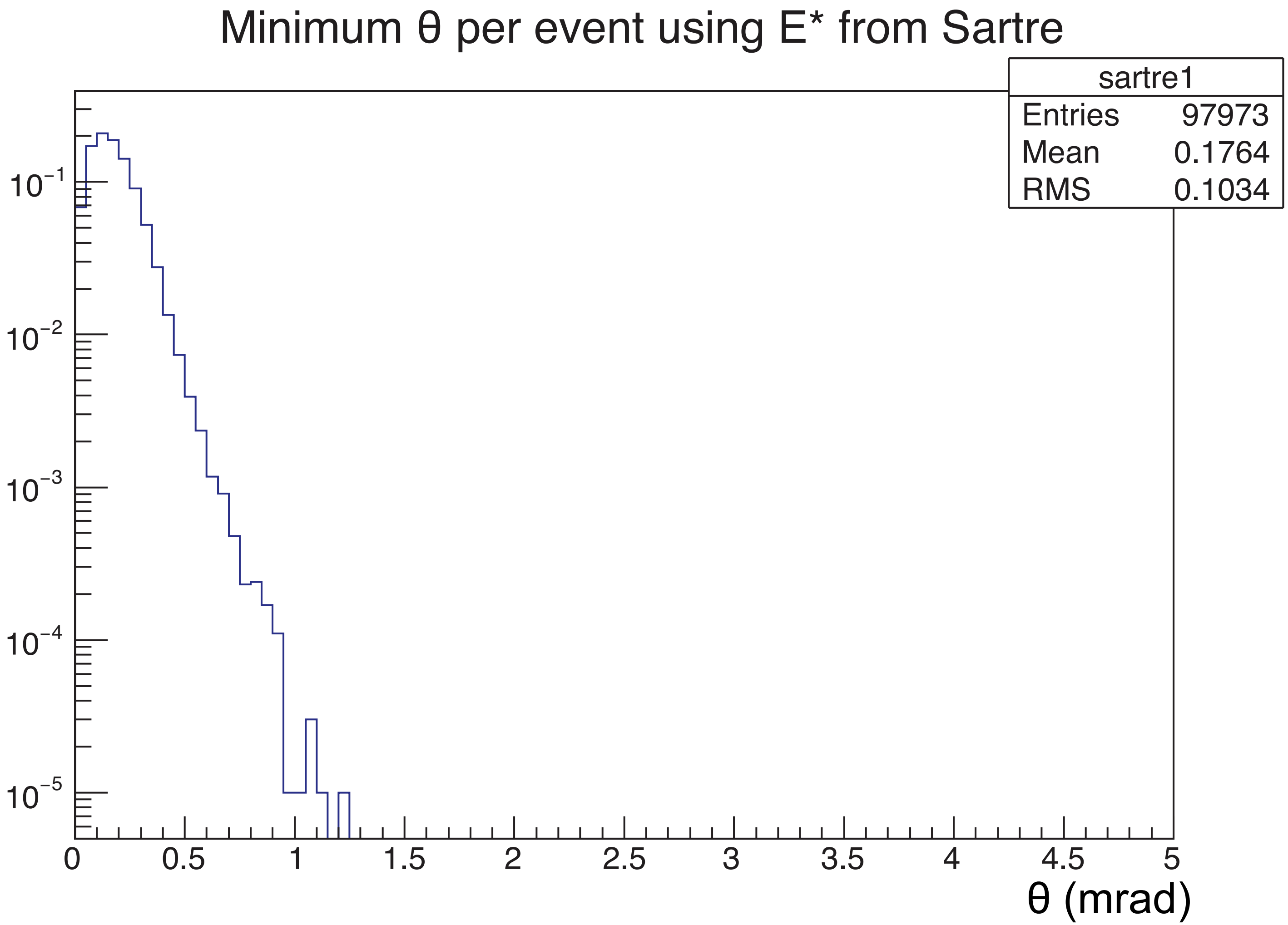
Results: # of Neutrons below given Δθ (II)



Results: Full Simulations (I)

Using E* distribution as generated by Sartre

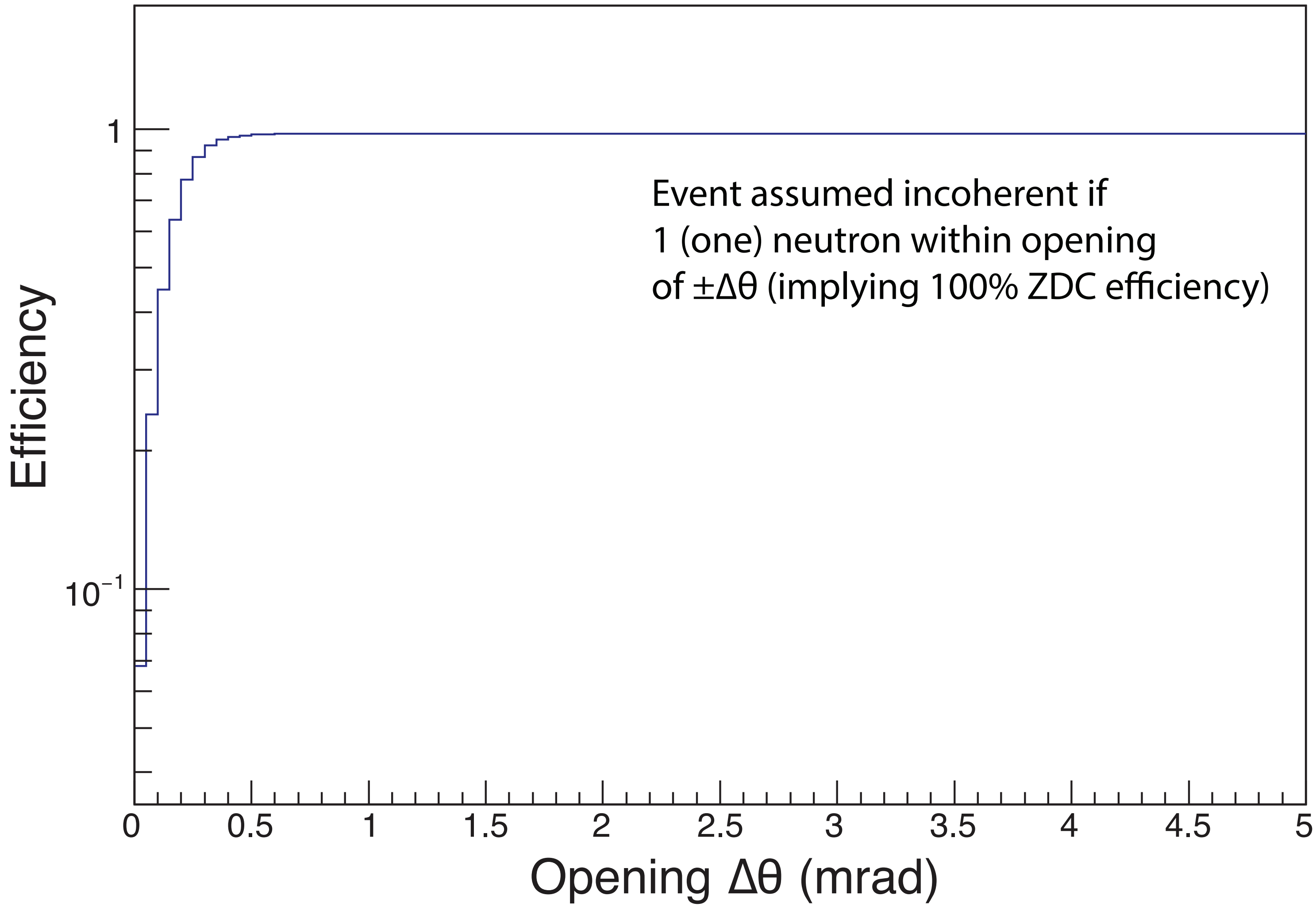
Here smallest θ of any neutron in each event



Results: Full Simulations (II)

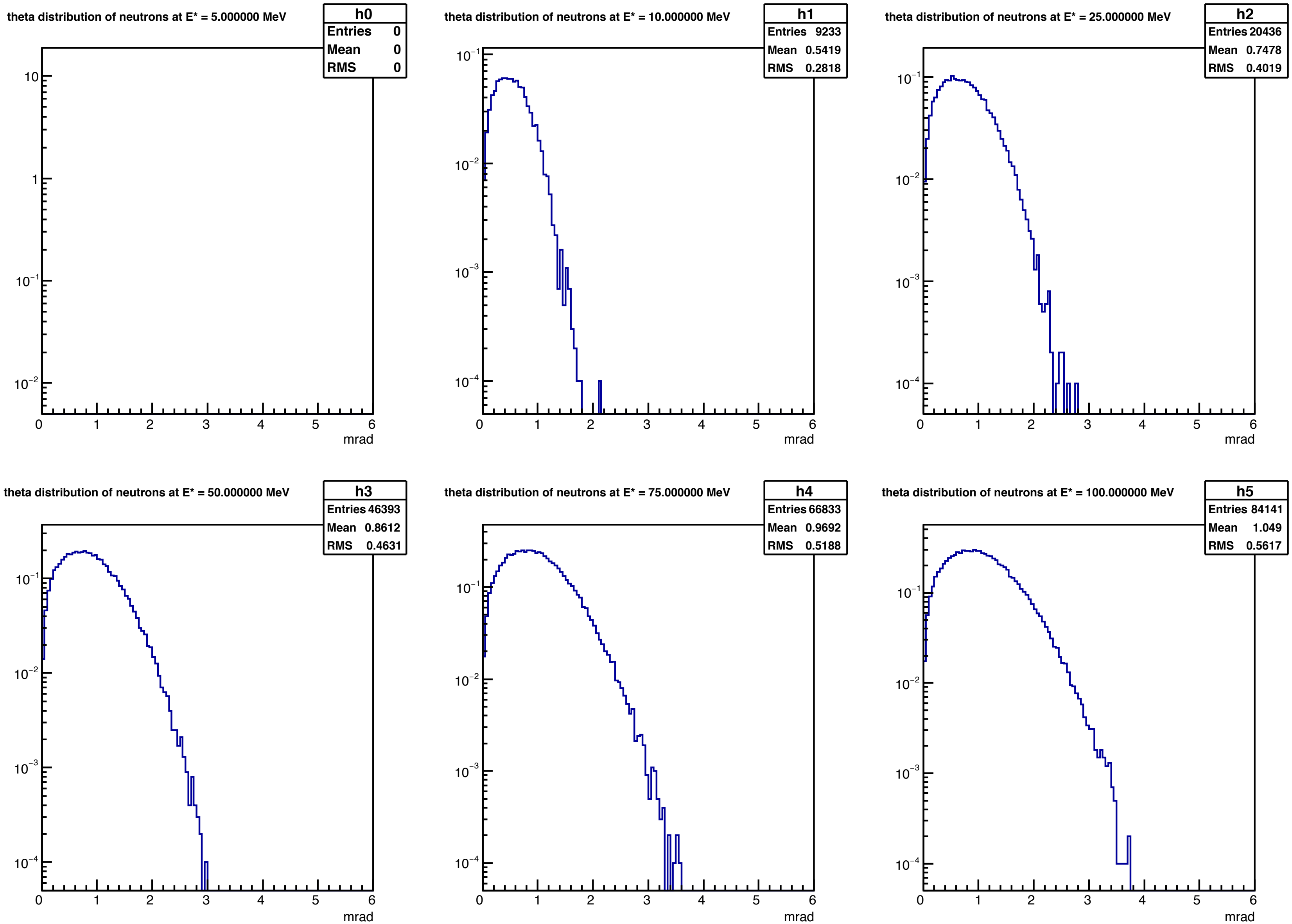
Using E* distribution as generated by Sartre

Incoherent (breakup) tagging efficiency

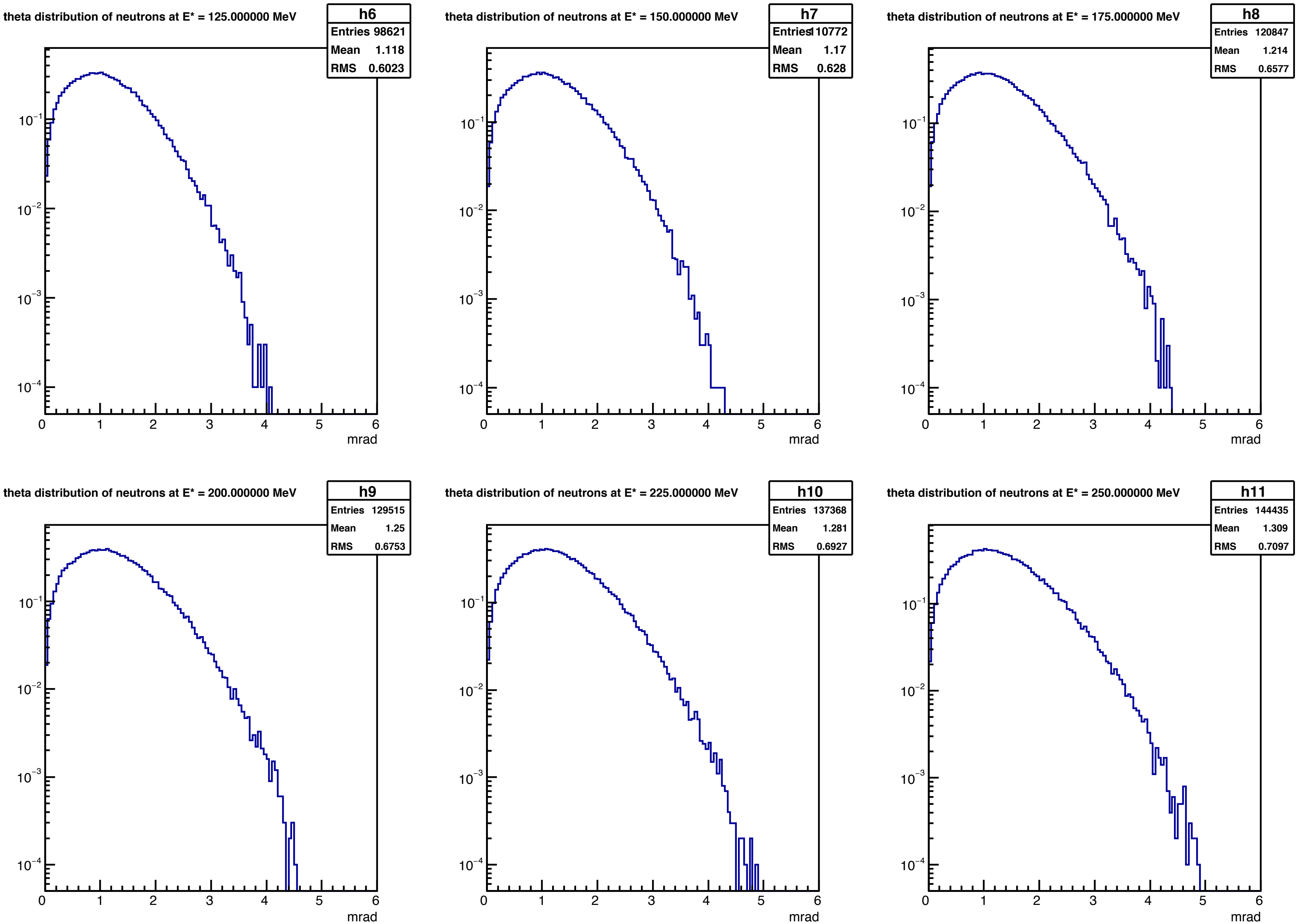


Au 50 GeV

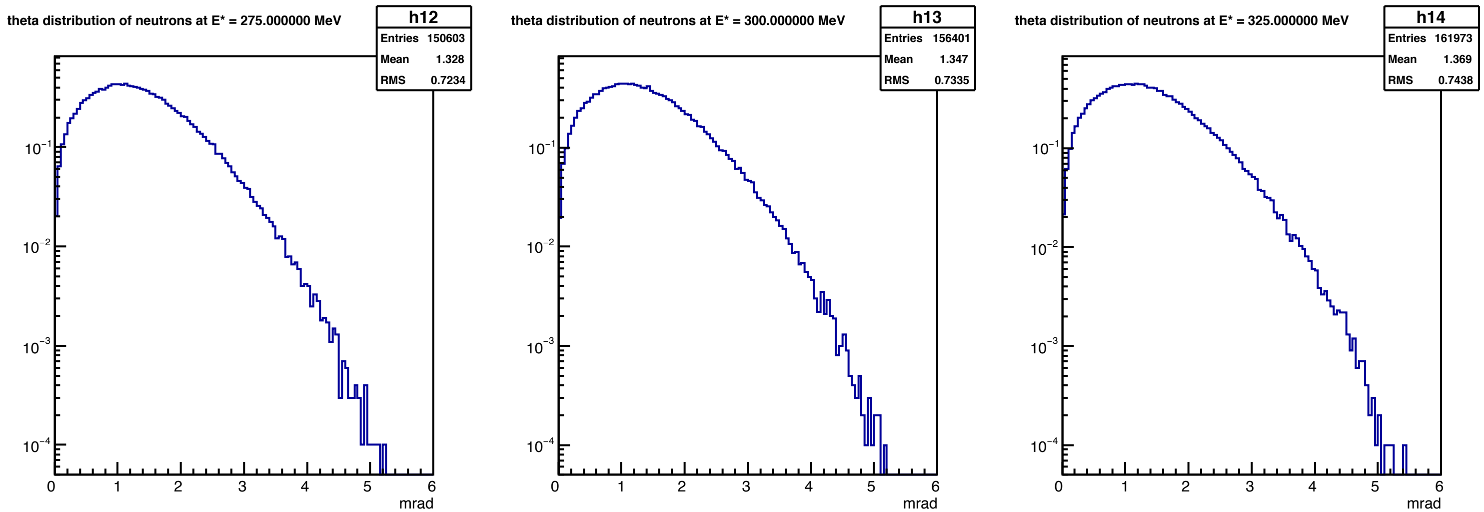
Results: θ Distribution of Neutrons (I)



Results: θ Distribution of Neutrons (II)



Results: θ Distribution of Neutrons (III)

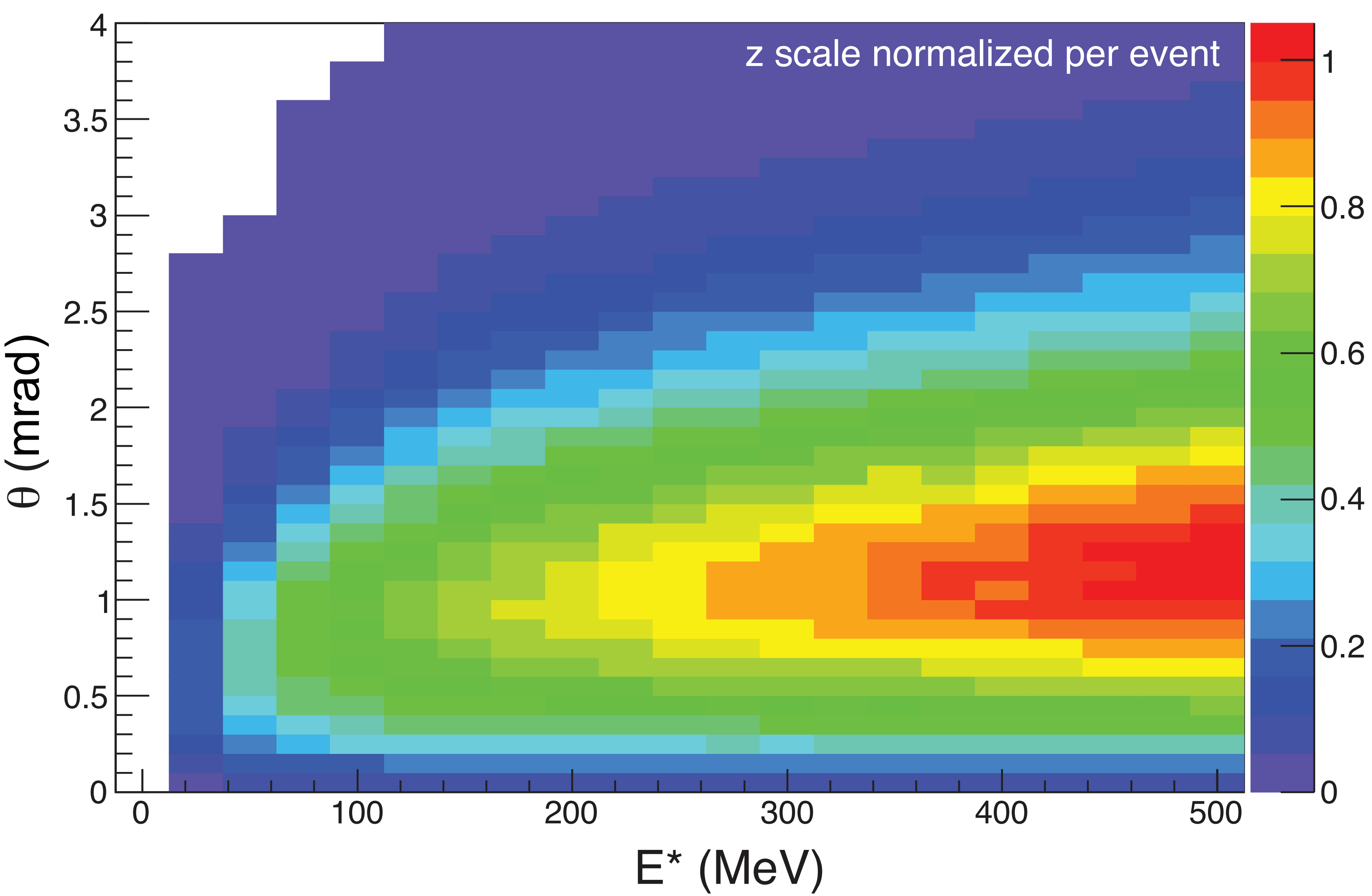


Note: Same as before, above E* = 100 MeV the Median (max) is not shifting

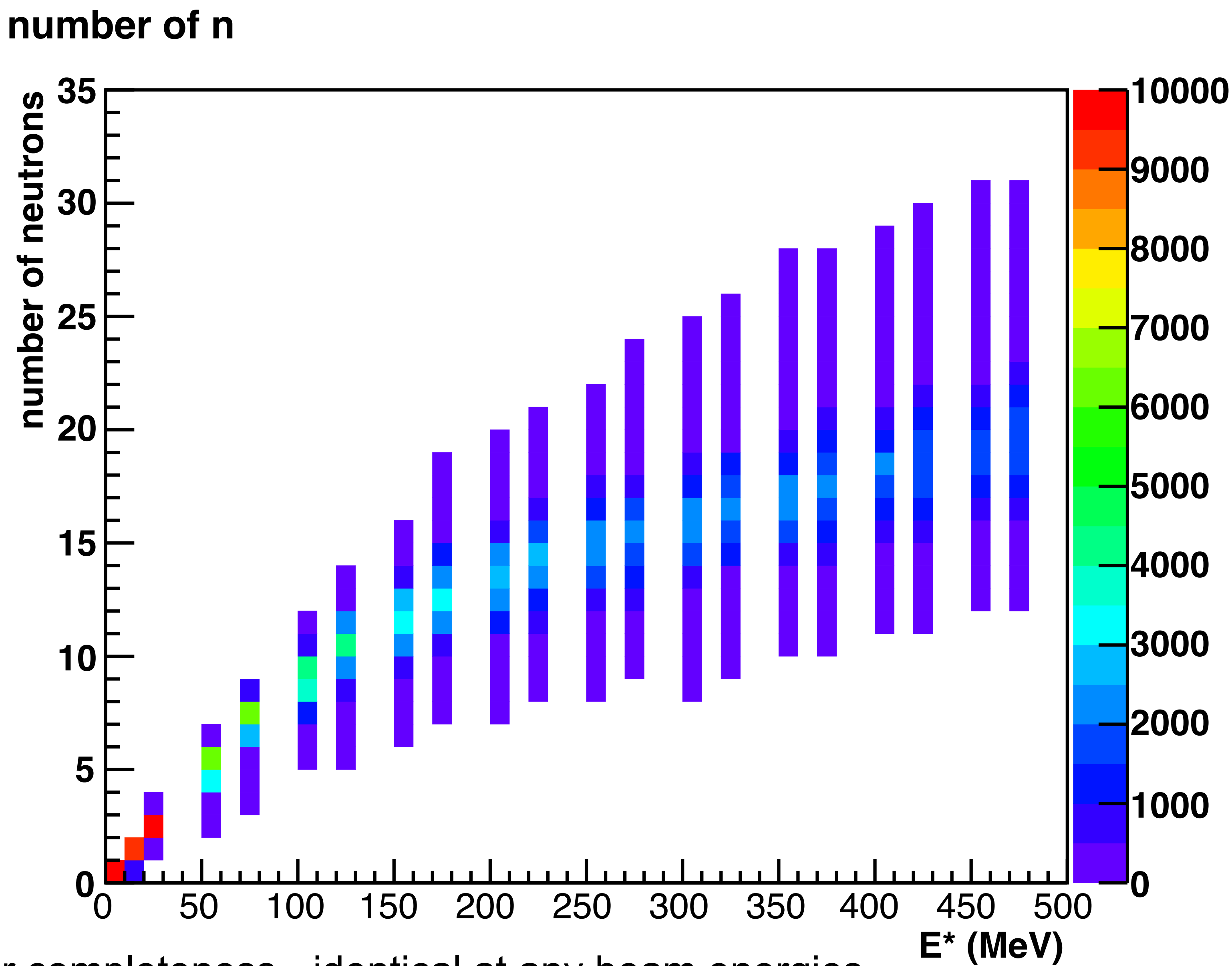
but

extends **much** further in θ

Results: θ vs E* (⟨per event distribution⟩)



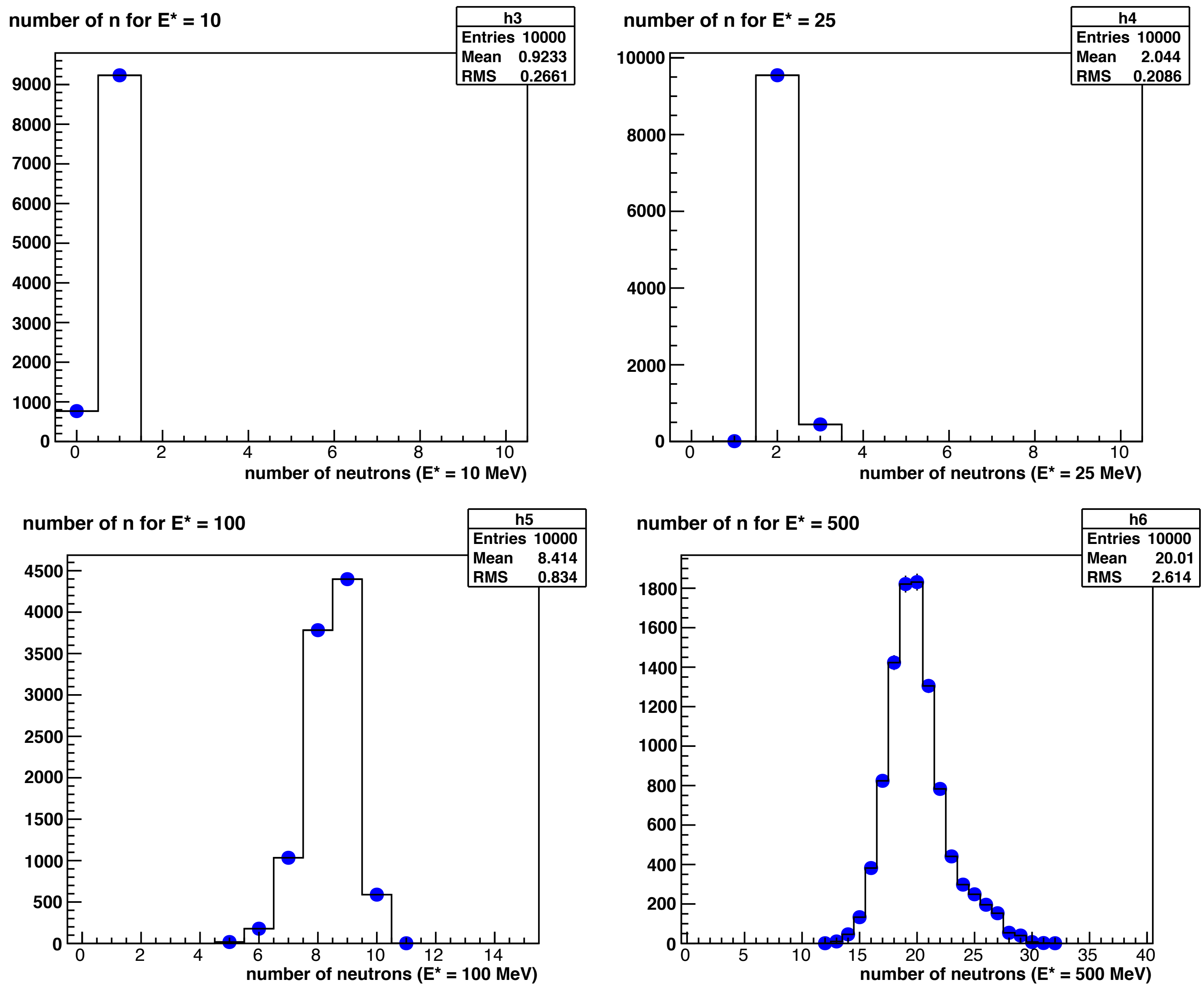
Results: Number of Neutrons (I)



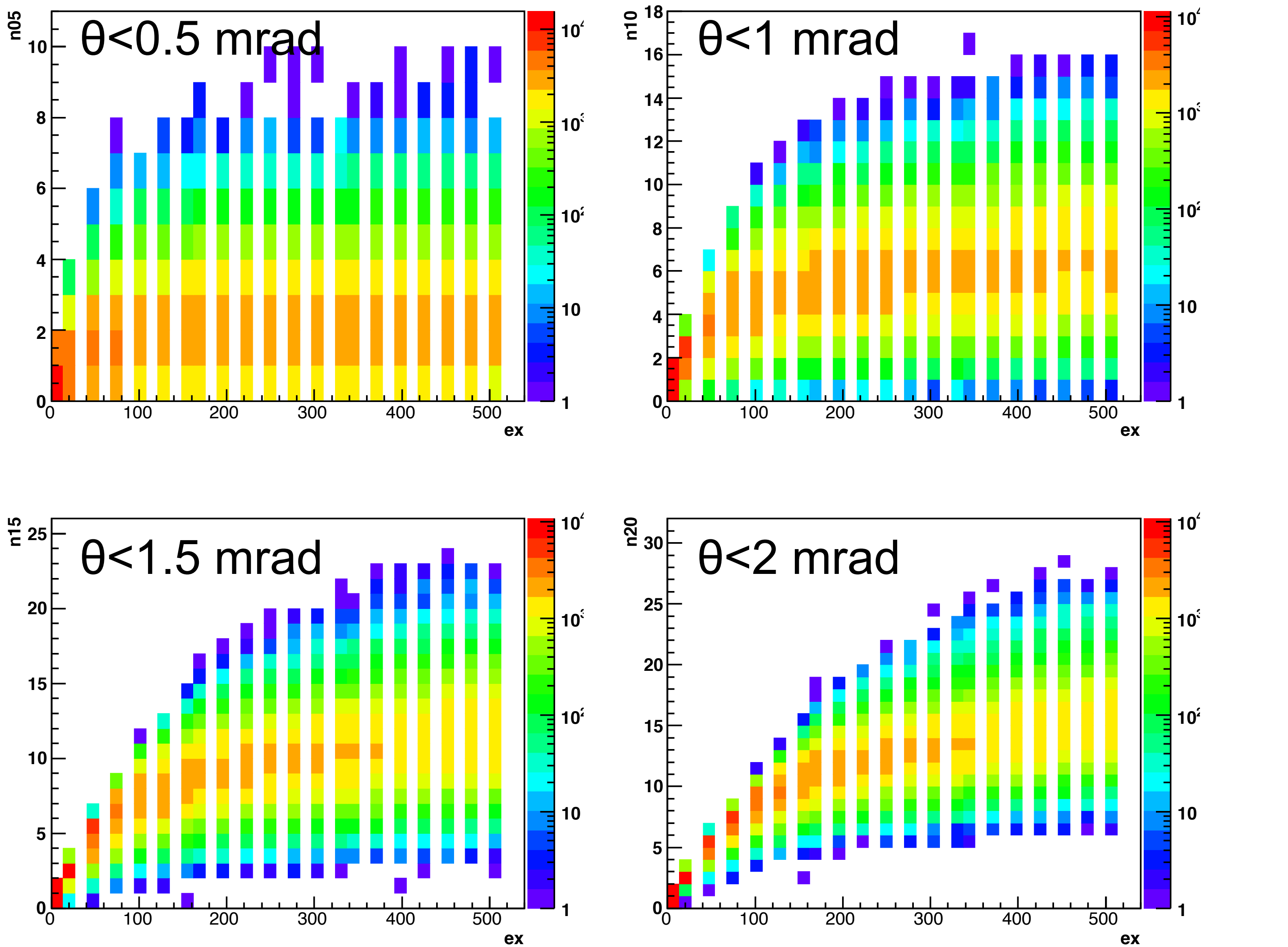
Just for completeness - identical at any beam energies

Results: Number of Neutrons (II)

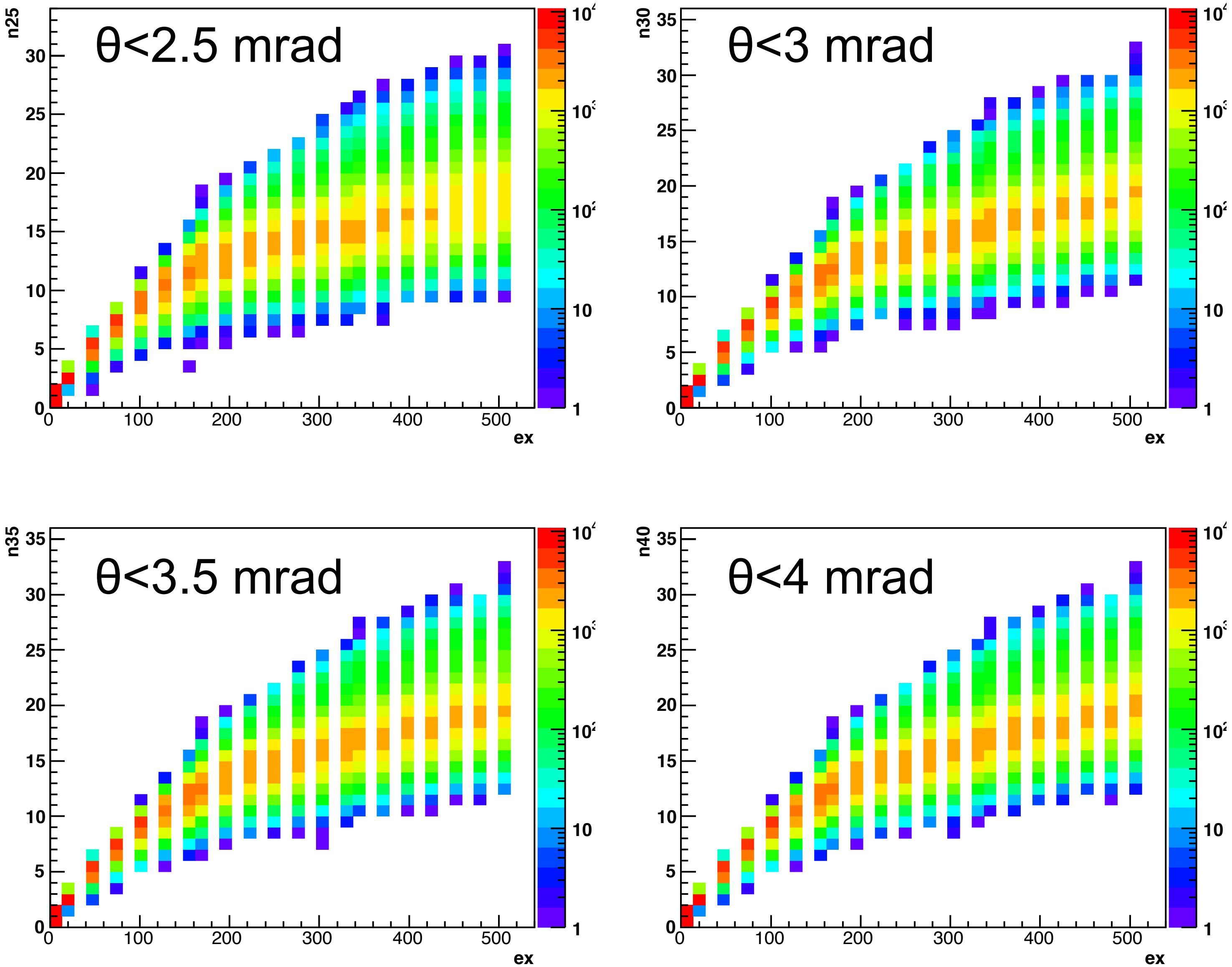
Just for completeness - identical at any beam energies



Results: # of Neutrons below given Δθ (I)



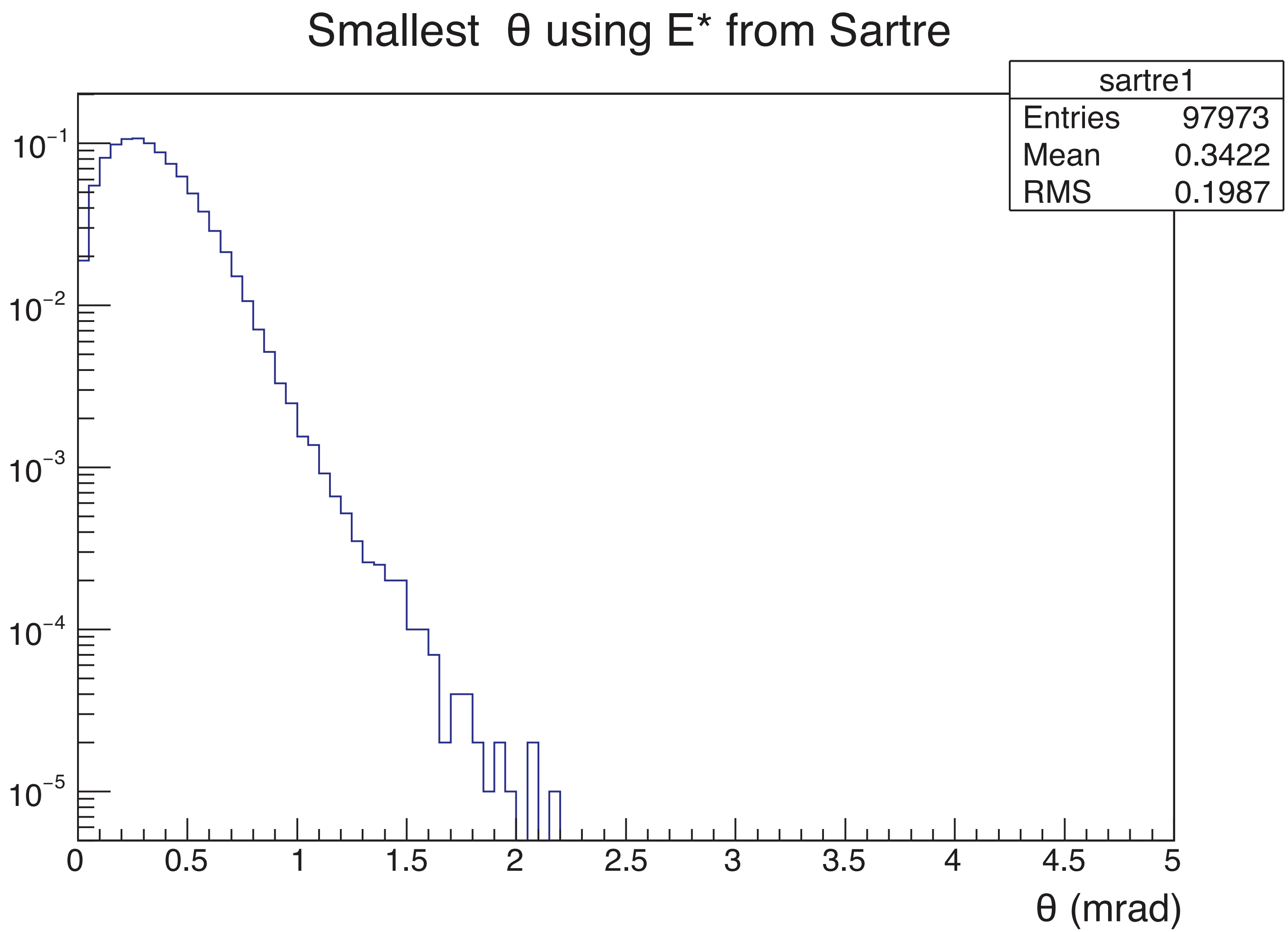
Results: # of Neutrons below given Δθ (II)



Results: Full Simulations (I)

Using E* distribution as generated by Sartre

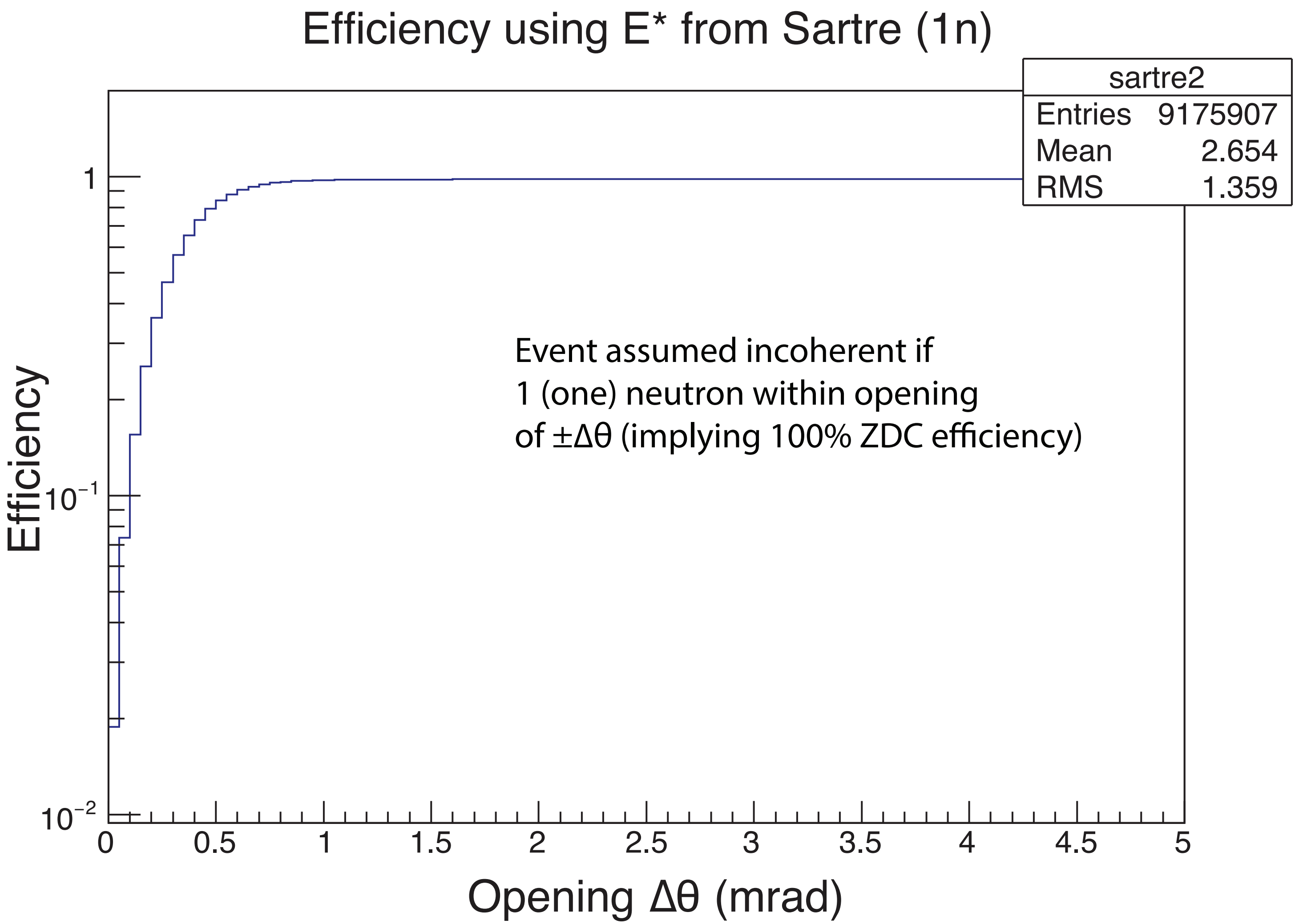
Here smallest θ of any neutron in each event



Results: Full Simulations (II)

Using E* distribution as generated by Sartre

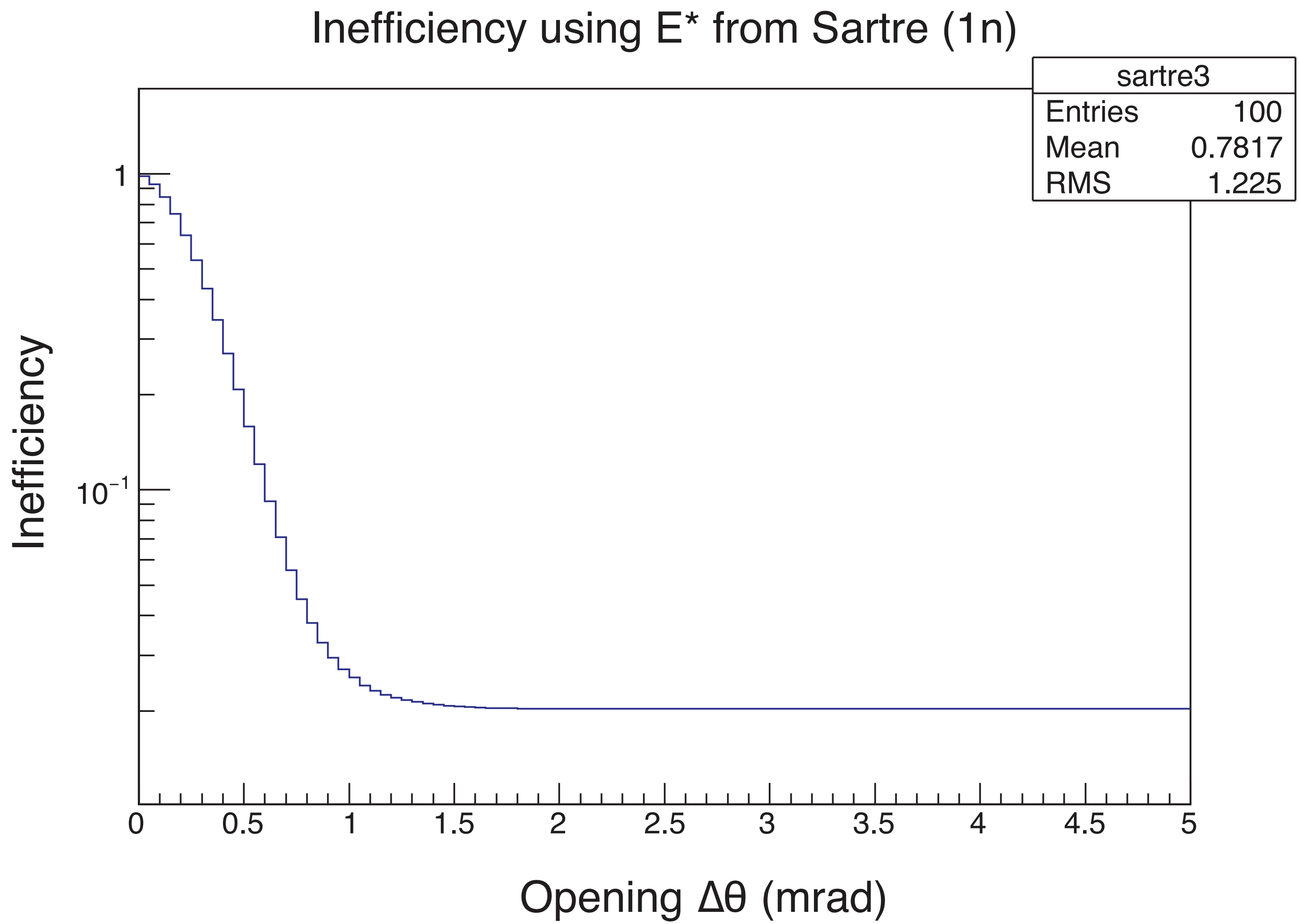
Incoherent (breakup) tagging efficiency



Results: Full Simulations (III)

Using E* distribution as generated by Sartre

Incoherent (breakup) tagging inefficiency (= 1-efficiency)



Al 100 GeV
(Z=13, A=27)